AFFDL-TR-67-144
Volume III

AD 667804

INVESTIGATION OF TURBULENT HEAT TRANSFER AT HYPERSONIC SPEEDS

Volume III. The Laminar-Turbulent $\rho_r \mu_r$ Momentum Integral and Turbulent Nonsimilar Boundary Layer Computer Programs

R. T. Savage
C. L. Jaeck
J. R. Mitchell

THE BOEING COMPANY

TECHNICAL REPORT AFFDL-TR-67-144, VOLUME III

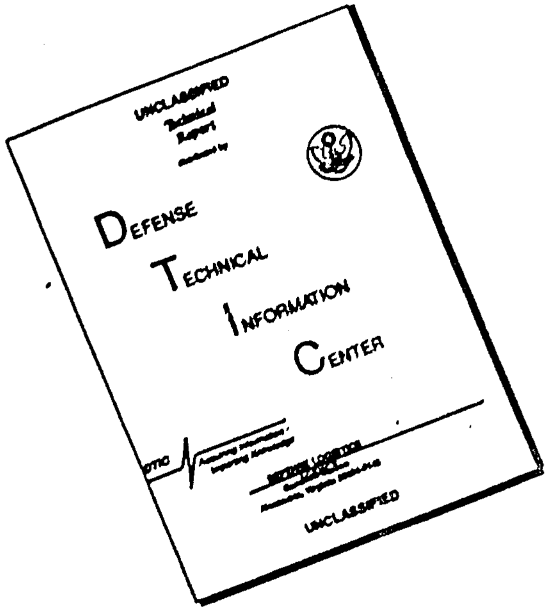
December 1967

This document has been approved for public release and sale; its distribution is unlimited.

Air Force Flight Dynamics Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

232

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

INVESTIGATION OF TURBULENT HEAT TRANSFER AT HYPERSONIC SPEEDS

Volume III. The Laminar-Turbulent $\rho_r \mu_r$ Momentum Integral and Turbulent Nonsimilar Boundary Layer Computer Programs

R. T. Savage
C. L. Jaeck
J. R. Mitchell

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned to the Air Force Flight Dynamics Laboratory unless return is required by security considerations, contractual obligation, or notice on a specific document.

FOREWORD

This report was prepared by the Space Division, Aerospace Group of The Boeing Company, Seattle, Washington, under direction of Messrs. A. L. Nagel and V. Deriugin, program managers. The contract was initiated under BPSN 5(611366-62405334), Project 1366 Hypersonic Gas Dynamic Heating, Task 136607 Aerodynamics and Flight Mechanics, USAF Contract AF33(615)-2372, Investigation of Turbulent Heat Transfer at Hypersonic Speeds. The work was administered by the Air Force Flight Dynamics Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. Mr. Richard D. Neumann (FDMG) was the Air Force project engineer.

Results obtained during this program are published in three volumes. Volume I, Analytical Methods; Volume II, Analysis of Heat Transfer and Pressure Data on a Flat Plate, Cone, Ogive, Cylindrical Leading Edge, Blunt Delta Wing, and X-15 Aircraft; and Volume III, The Laminar-Turbulent [illegible] Momentum Integral and Turbulent Nonsimilar Boundary Layer Computer Programs. Boeing document numbers assigned to these volumes are D2-113531-1, -2, and -3, respectively.

This report covers work conducted between March 1965 and March 1967. The report was submitted by the authors in May 1967.

The authors acknowledge Barbara J. Safley for her exceptional effort in generating and editing the figures contained in this report.

This technical report has been reviewed and is approved.

Philip P. Antonatos

PHILIP P. ANTONATOS
Chief, Flight Mechanics Division
Air Force Flight Dynamics Laboratory

ABSTRACT

This report presents a combined analytical and experimental investigation of turbulent heat transfer on basic and composite configurations at hypersonic speeds. The analytical results are presented in Volume I, the experimental results, including data-theory comparisons, are presented in Volume II, and computer programs incorporating the analytical methods described herein are presented in Volume III.

The two heat-transfer prediction methods programmed are the laminar-turbulent $\rho_r \mu_r$ momentum integral method and the turbulent nonsimilar boundary layer method. This volume of the report describes the numerical method and presents flow charts, program listings, input forms, and a description of the output for each program. The programs are written in Fortran IV language for operation on the IBM 7094.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

Two methods for the calculation of turbulent heating rates are discussed in Volume I of this report, the laminar-turbulent momentum integral method ($\rho_r \mu_r$) and the nonsimilar method for turbulent boundary layers (NSBL).

This volume describes the computer program for each method. A description of the numerical method is given as well as flow charts, program listings, input preparation, and output description.

The first computer program was written incorporating the $\rho_r \mu_r$ heat transfer prediction method developed by R. A. Hanks of The Boeing Company. The method was developed in the course of the X-20 program and finalized under NASA Research Contract No. NAS 8-11321 (Reference 1). In addition to the $\rho_r \mu_r$ method calculations, the program also contains subprograms for calculating pressures, gas properties, and streamline patterns for several special cases. The SRU 1108 version of the program was written during Boeing Company-funded research and converted to the IBM 7094 during this Air Force Flight Dynamics-funded study.

The program has four major sections. Program A contains the method per se. Given the external flow properties and wall condition, this section of the program computes laminar or turbulent heating rates. All required gas properties are computed by the program from the given pressure and edge velocity. The effect of transition is estimated by matching laminar and turbulent boundary layer momentum thicknesses at the transition point. The point of transition is determined by the program on the basis of a transition Reynolds number selected by the user. Program A can be applied to any geometry for which the external flow properties including nonisothermal wall effects can be defined by the user.

The second section, Program B, computes the local velocities and static temperatures along a streamline from a given pressure distribution.

If the streamline divergence is not known, Program C can be used to obtain this information. Program C calculates the path of two streamlines, the streamline of interest and one below, given a two-dimensional array of streamline angles. With two streamline paths known, the divergence parameter can be computed. This calculation, however, assumes zero divergence due to body geometry.

The fourth section of the computer program, called D, calculates or provides information for the three previous subprograms for four special cases; the axisymmetric and two-dimensional body, the hemisphere, the swept cylinder, and the sharp delta wing.

The second computer program, the turbulent nonsimilar boundary layer, integrates the boundary layer partial differential equations (described in Appendix C of Volume I) using finite difference methods. The program can calculate the turbulent boundary layer on unyawed sharp or blunt axisymmetric or two-dimensional bodies. Three-dimensional flow effects are calculated using the zero crossflow pressure gradient approximation which implies no rotation of the velocity vectors within the boundary layer. The program is also limited to attached flow in air, which is considered as ideal gas.

The program is capable of initiating its own boundary layer solutions, given only external flow properties, for the stagnation point of either sharp or blunt tip cones and plates.

Special inputs required are: pressure, wall enthalpy, a three-dimensional flow parameter and its derivative, the velocity gradient at the boundary layer edge and the normal velocity at the surface, all functions of the streamwise distance x. The user must also specify an initial and final value of x, the value of the x increments, and printout information.

Output includes streamwise and normal velocity, temperature, total and static enthalpy, shear, heat transfer rate, a similarity parameter, and x derivatives of velocity and total enthalpy. These values are tabulated as functions of y (normal to surface) at each output position. Also printed in the output are the shear and heat transfer rate at the wall, and the boundary layer displacement and momentum thickness.

The $\rho_r \mu_r$ program requires about 0.5 to 1 minute of computer time per case, while the nonsimilar program requires 2 to 3 minutes for a flat plate case. The time estimate can vary and depends largely on the type of case.

SECTION II

THE LAMINAR-TURBULENT MOMENTUM INTEGRAL COMPUTER PROGRAM ($\rho_r\mu_r$)

1. DISCUSSION

The computer program described in this report is based on the Hanks $\rho_r\mu_r$ method of solution to the boundary layer momentum integral equation; and, as such, contains all of the assumptions pertaining to the method as discussed in References 1, 2, 3, and Volume I. The program calculates aerodynamic heating on arbitrary bodies, with or without streamwise pressure gradients in laminar or turbulent flow. Three-dimensional flow effects are included in the form of streamline divergence due to body geometry (r) and crossflow or transverse pressure gradients (f). The effect of a nonisothermal wall on aerodynamic heating is also calculated with modified methods of Lighthill and Seban. This is discussed in further detail in Reference 1. The program described herein is applicable only to air in chemical equilibrium (References 4-7).

a. Method of Solution

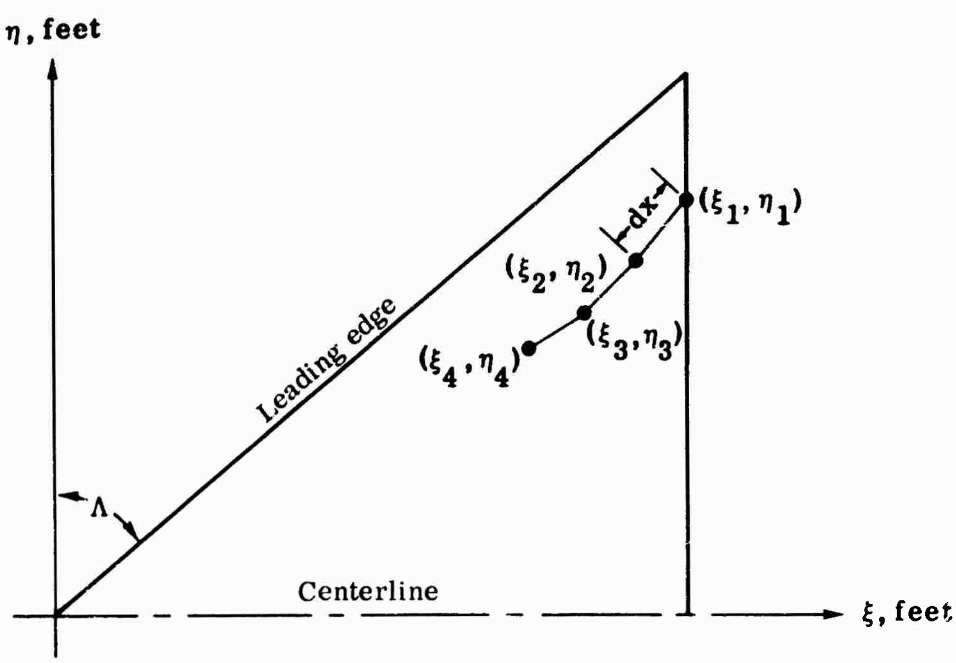
The $\rho_r\mu_r$ integral program is actually four programs, any of which can be operated separately. The first program, Program A, calculates heating rates for any body shape at any flight or ground-facility test condition given the external flow properties, gas properties at the wall, gas properties at the stagnation condition, total streamline divergence and the streamline divergence due to body geometry alone.

Program A has an additional capability of calculating transition effects based on a transition Reynolds number selected by the user. The program matches laminar and turbulent boundary layer momentum thicknesses at the transition point.

In addition, the program calculates reference heat transfer coefficients which are used to normalize the calculated local heat transfer coefficients. The laminar heat transfer coefficients are all normalized by the value of the hemisphere stagnation-point heat transfer coefficient evaluated at the same free stream conditions as the body of interest. The heat transfer coefficient on the stagnation line of a 60° swept infinite cylinder is used to normalize all turbulent heat transfer coefficients. The hemisphere and cylinder radius are items of input.

Program B calculates the boundary edge velocity and temperature given the pressure distribution along the streamline or body.

Program C computes streamwise pressure and divergence parameters given a two-dimensional field of pressures and streamline angles. Since pressure data and streamline angles are often most easily available as spanwise plots at various stations, considerable crossplotting is required to provide specific values along the streamline passing over any specified point on the body. This work is performed by Program C, which begins at the point for which the user desires to calculate the heating rate, and traces out the streamline, proceeding upstream to a specified boundary (e.g., the boundary layer origin). The streamline angle is determined by double interpolation of a two-dimensional array of angles which is input. The streamline angle is then used to determine the coordinates of a point on the streamline a distance dx upstream. This procedure is repeated until the upstream boundary is reached.



It also interpolates as required to obtain initial values at the upstream boundary of the input array (ξ, η) thus providing all information required for Program B.

Program D provides the pressure and streamline arrays and initial values required by Programs A, B, and C for several specific bodies. These optional configurations are:

D-1 Arbitrary two-dimensional bodies at an angle of attack and arbitrary axisymmetric bodies at zero angle of attack. The bodies may be sharp or blunt and composed of wedges and plates or cones and cylinders.

D-2 Hemisphere

D-3 Swept infinite cylinder ($0 < \Lambda < 90°$)

D-4 Sharp delta wing ($60° < \Lambda < 80°$ and $\alpha > 0$)

For options D-1 and D-2, the user provides a table of body width or radius at various locations along the axis. For options D-2 and D-3, the cylinder or hemisphere diameter must be input and for options D-3 and D-4, the cylinder or delta wing sweep angle must be provided. For option D-4 the angle of attack must be stated. In all options, the user provides the free stream conditions.

Options D-2 and D-3 (hemisphere and swept infinite cylinder) use pressure distributions stored within the program. These distributions are shown in Figure 1. Options D-1 and D-4 calculate pressures from the local flow deflection angle and a modified Newtonian pressure expression. For $\delta > 0$

$$\frac{C_p}{\sin^2 \delta} = 1.05 + \left[1.1025 - \frac{4}{(M_\infty \sin \delta)^2}\right]^{1/2} - \frac{1.278 \sin^2 \delta}{M_\infty^{0.6}} \tag{1}$$

and for $\delta \leq 0$

$$\frac{C_p}{\sin^2 \delta} = 0 \tag{2}$$

This approximate analytical method was devised for predicting pressures from the Newtonian relationship in which K is allowed to vary.

$$C_p = K \sin^2 \delta \tag{3}$$

The relationship for K was chosen to obtain the best agreement with the exact solutions for a wedge, cone, and blunt body stagnation point. A plot of the modified pressure coefficients is shown in Figure 2.

The D options also provide velocity and streamline angle distributions along a streamline. In option D-1, the velocity is computed from

$$u_e = u_\infty \left[1 - \left(\frac{P}{P_\infty} - 1\right) \frac{1}{\gamma M_\infty^2}\right]^{1/2} \quad \text{if } P_{initial} \leq .528\, P_o \tag{4}$$

and

$$u_e = a^* \frac{x}{x_{sonic\ point}} \quad \text{if } P_{initial} > .528\, P_o, \quad x < x_{sonic\ point} \tag{5}$$

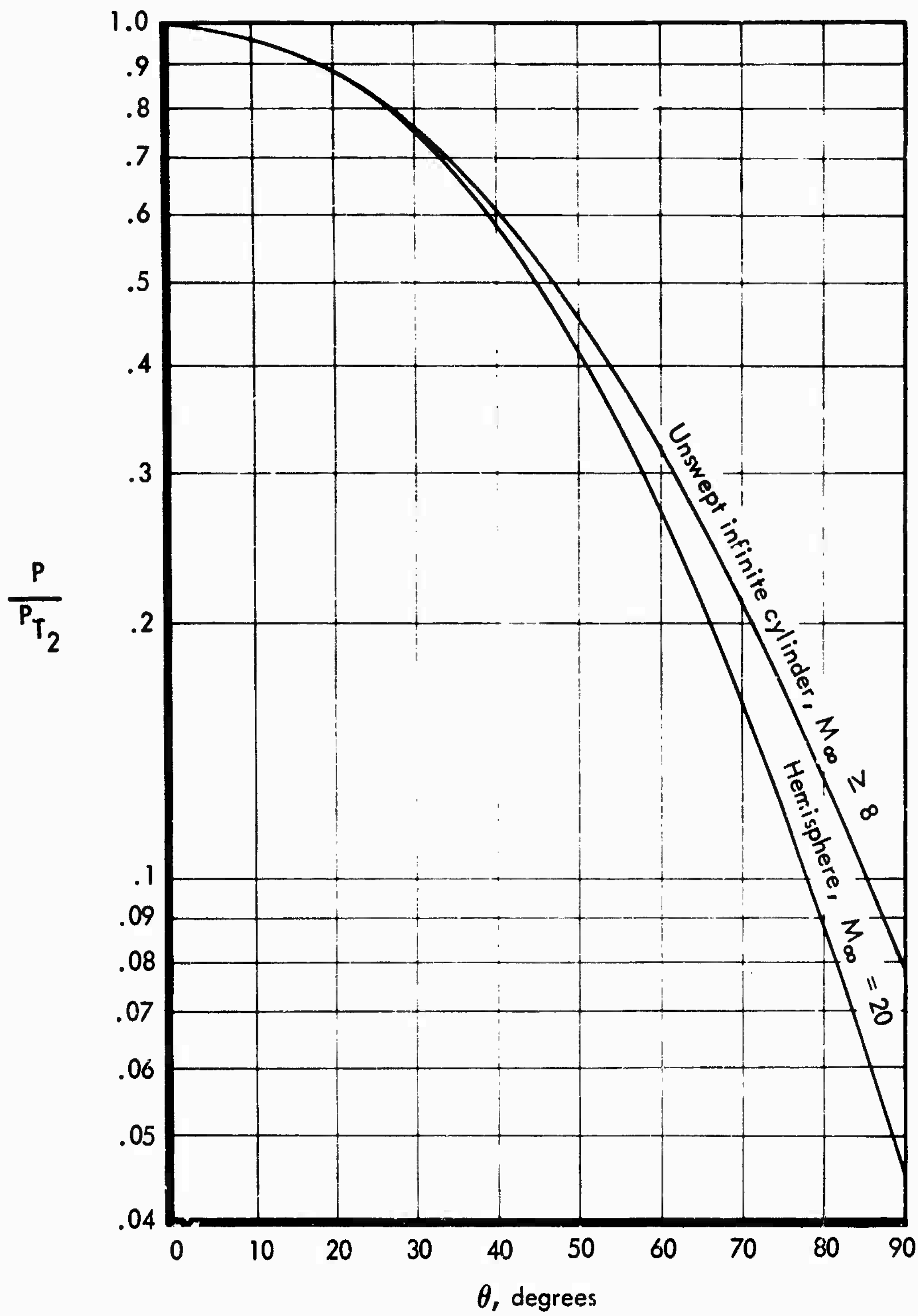


Figure 1: PRESSURE DISTRIBUTION ON A HEMISPHERE AND UNSWEPT INFINITE CYLINDER (REF. 3)

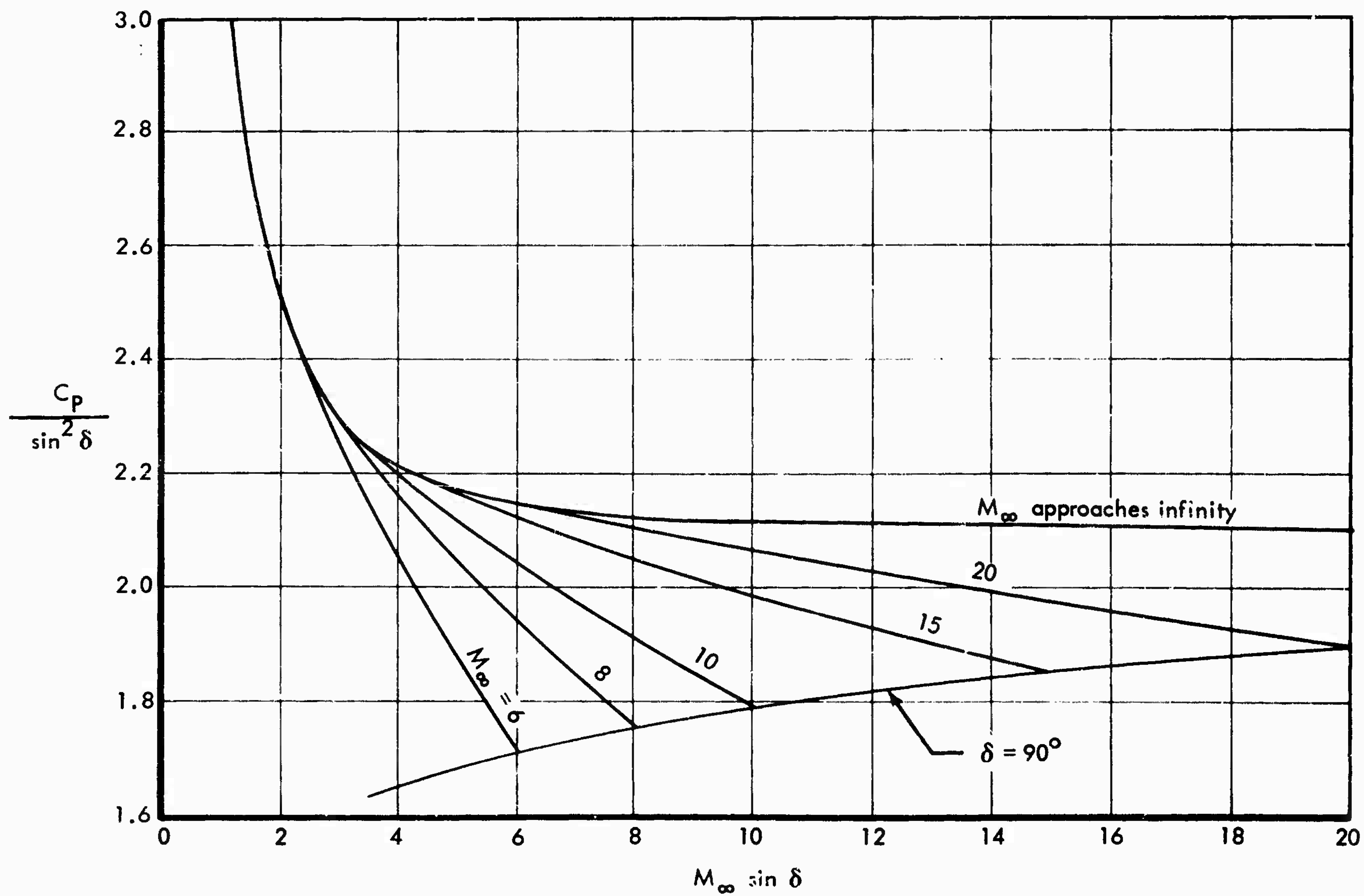


Figure 2: BOEING MODIFIED NEWTONIAN HYPERSONIC PRESSURE COEFFICIENTS

and

$$u_e = a^* \left\{ 6 \left[1 - \left(\frac{P}{P_o} \right)^{2/7} \right] \right\}^{1/2} \qquad x > x_{\text{sonic point}} \tag{6}$$

where a^* is the velocity at the sonic point

Program options D-2 and D-3 obtain the edge velocity by integration of the pressure gradient along the streamline, while D-4 calculates u_e from

$$u_e = u_\infty \left[1 - \frac{\delta^2}{5600} \right] \tag{7}$$

The development of Equation (7) is presented in detail in Reference 3.

In addition, options D-3 and D-4 for the swept cylinder and delta wing also provide a two-dimensional array of streamline angles. The streamlines angles on the swept cylinder are obtained from

$$\tan \theta_e = \frac{v_e}{u_\infty \sin \Lambda} \tag{8}$$

where the spanwise velocity v_e is obtained by integration of the pressure gradient normal to the leading edge. Delta wing streamline angles are obtained from a correlation curve and

$$\theta_e = (90 - \Lambda) \left(\frac{d\theta}{d\epsilon} \right)_{CL} \bar{R} + \left[\theta_{LE} - (90 - \Lambda) \left(\frac{d\theta}{d\epsilon} \right)_{CL} \right] \bar{R}^9 \tag{9}$$

where

$$\bar{R} = \frac{\tan \left(\frac{\eta}{\xi} \right)}{\tan \left(\frac{90 - \Lambda}{57.3} \right)}$$

$\left(\frac{d\theta}{d\epsilon} \right)_{CL}$ is the gradient of the streamline angle with respect to the ray angle ϵ

The streamline angle distribution across the wing span is a curve fit obtained by matching the boundary conditions given by the angle and the gradient at the centerline and the angle at the leading edge. The calculations of program D are necessarily somewhat approximate since exact flow field calculations are possible for only a few simple geometries. The approximations used in program D are those discussed in Reference 1 and are simple yet reasonably accurate.

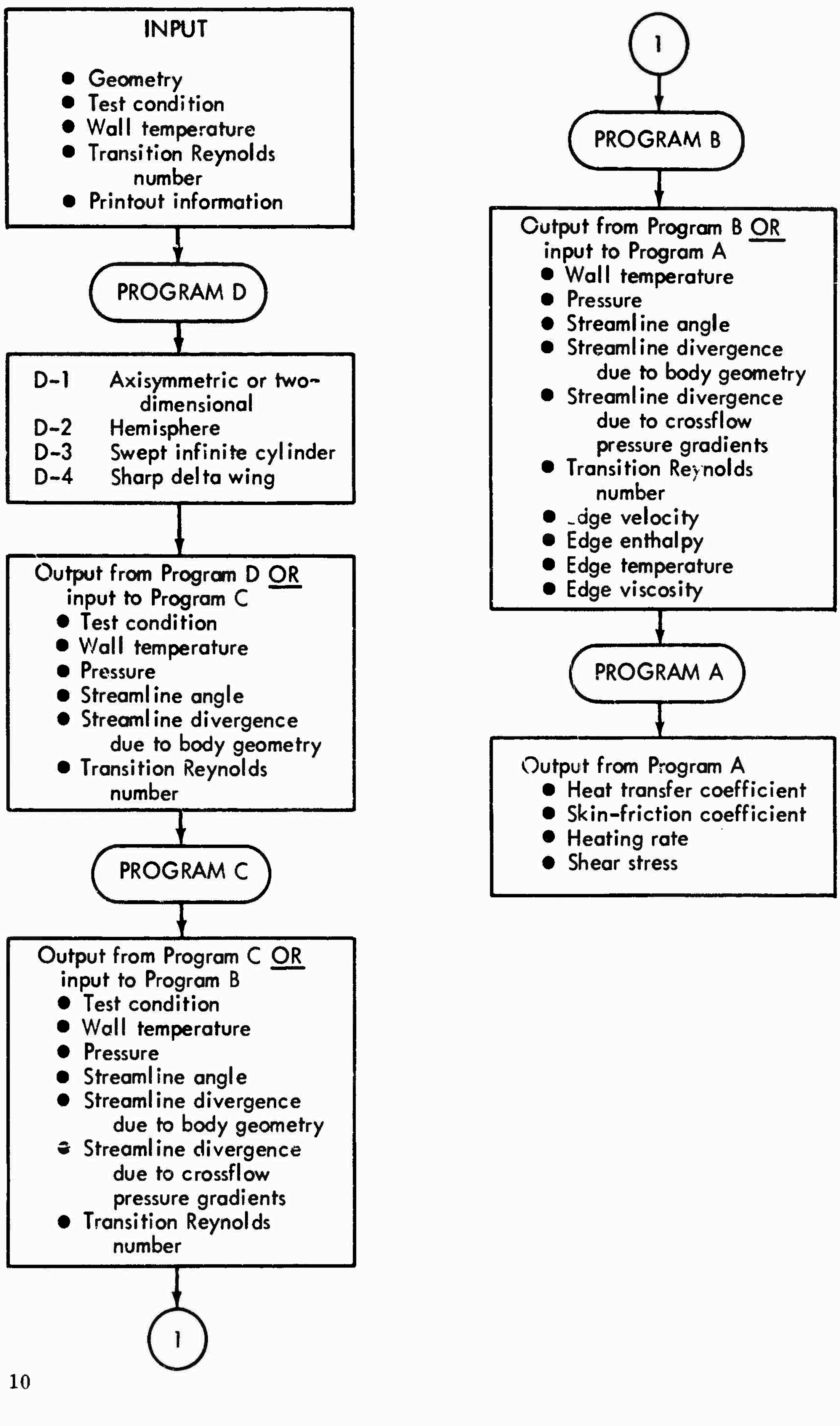
Program D is easily modified and extended as more information becomes available, or as certain specific shapes become important.

b. $\rho_r \mu_r$ Program Schematic

The relation of the four programs is illustrated on the following page: however, it should be understood that all four programs are coupled and function together as a single program when desired. The symbol OR indicates that the required data can come from either of the indicated sources.

c. Nomenclature

a^*	sonic velocity
C_p	pressure coefficient
f	streamline divergence due to crossflow pressure gradients
M_∞	free stream Mach number
P	local pressure
P_o	stagnation point pressure
P_{T_2}	stagnation point or stagnation line pressure
r	streamline divergence due to body geometry
u_e	boundary layer edge velocity
u_∞	free stream velocity
v_e	spanwise velocity at the boundary layer edge
x	distance along a streamline
δ	local flow deflection angle measured from the free stream velocity vector
Δ	total streamline divergence, includes body geometry and crossflow pressure gradients (rf)
ϵ	ray angle on a delta wing
θ	angular location


INPUT
● Geometry
● Test condition
● Wall temperature
● Transition Reynolds number
● Printout information
PROGRAM D
D-1 Axisymmetric or two-dimensional
D-2 Hemisphere
D-3 Swept infinite cylinder
D-4 Sharp delta wing
Output from Program D OR input to Program C
● Test condition
● Wall temperature
● Pressure
● Streamline angle
● Streamline divergence due to body geometry
● Transition Reynolds number
PROGRAM C
Output from Program C OR input to Program B
● Test condition
● Wall temperature
● Pressure
● Streamline angle
● Streamline divergence due to body geometry
● Streamline divergence due to crossflow pressure gradients
● Transition Reynolds number
1
1
PROGRAM B
Output from Program B OR input to Program A
● Wall temperature
● Pressure
● Streamline angle
● Streamline divergence due to body geometry
● Streamline divergence due to crossflow pressure gradients
● Transition Reynolds number
● _dge velocity
● Edge enthalpy
● Edge temperature
● Edge viscosity
PROGRAM A
Output from Program A
● Heat transfer coefficient
● Skin-friction coefficient
● Heating rate
● Shear stress

θ_e local streamline angle

Λ leading edge sweep angle

(ξ, η) coordinate system

$\rho_r \mu_r$ density-viscosity product evaluated at a reference condition

2. $\rho_r \mu_r$ PROGRAM DESCRIPTION

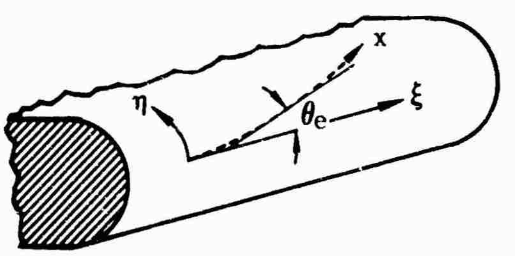
Presented in the following section is a description of the program and subroutines. Equations, nomenclature input, and output from each subroutine are presented in alphabetical order by the subroutine title.

Before proceeding, a description of the program coordinate system is required. The program operates in a two-dimensional curvilinear coordinate system where the coordinate axes are designated as η and ξ. Restrictions on this coordinate system are:

1. η and ξ are orthogonal.
2. Both η and ξ must be parallel to the surface over which the boundary layer flow is being considered, and
3. ξ is taken to be in the general direction of the fluid flow.

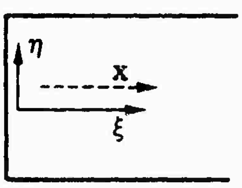
Thus, no coordinate axis projects outward or normal from the surface area. In the case of a flat surface, η and ξ will lie in the same plane.

CYLINDER

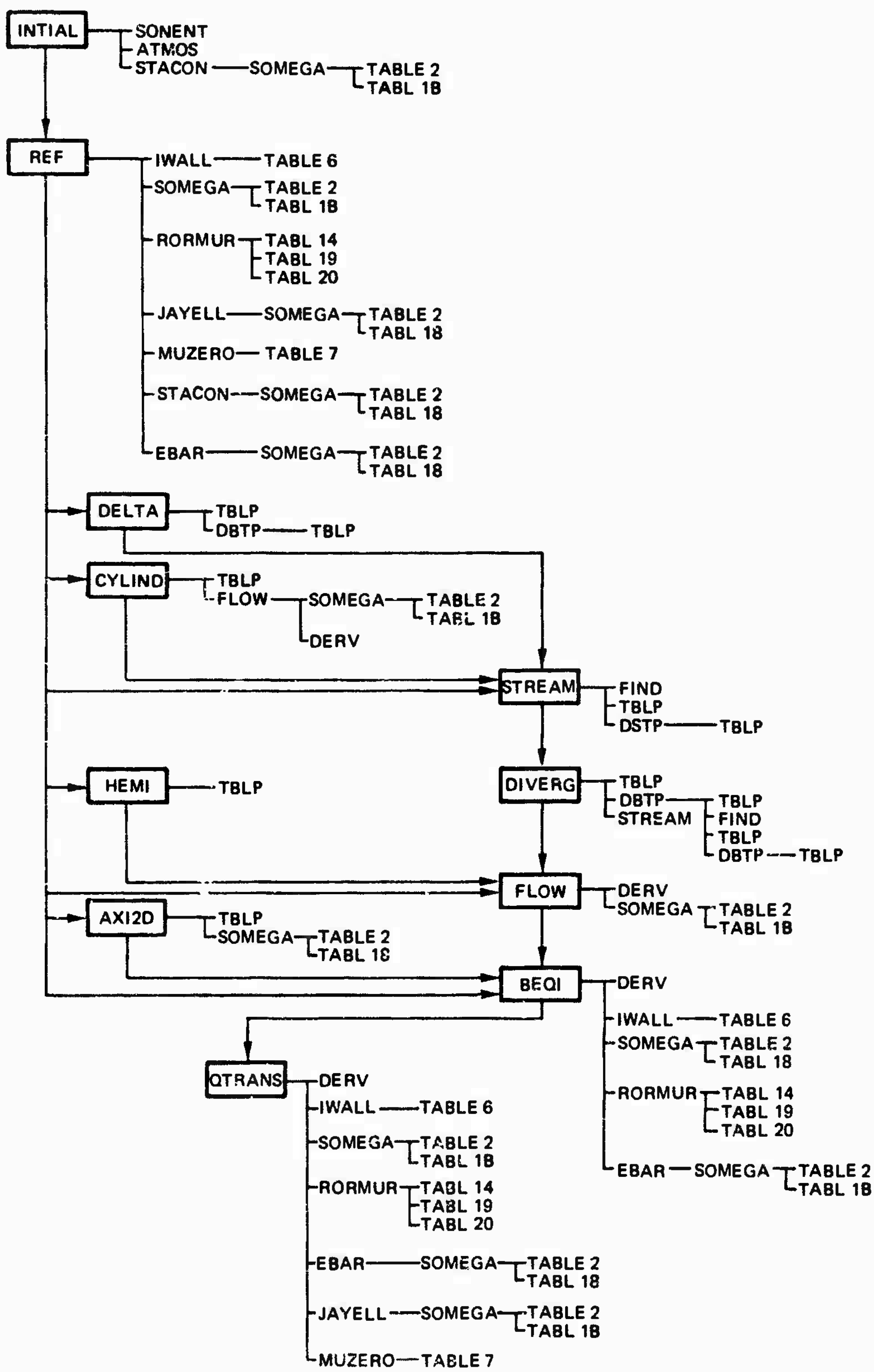


x, streamline distance

FLAT PLATE



a. $\rho_r \mu_r$ Program Flow Chart



b. Subroutines

The main program of the laminar-turbulent $\rho_r \mu_r$ momentum integral computer program — II (AS2419) — controls the input and the logic flow to the required mathematical subroutines. The input logic is arranged to minimize the modification required to change the input formats or the source of inputs (e.g., tape input or direct coupling to another program).

1) ATMOS (ALT, SONIC, PINF, RHO, TEMP)

Given the altitude, ATMOS calculates the free stream conditions. This subroutine is program AS1772, "Standard Atmosphere Properties, 1962".

Input

ALT		altitude, ft

Output

SONIC	a_∞	free-stream velocity of sound, ft/sec
PINF	P_∞	free-stream pressure, lb/ft^2
RHO	ρ_∞	free-stream density, slug/ft^3
TEMP	T_∞	free-stream temperature, °R

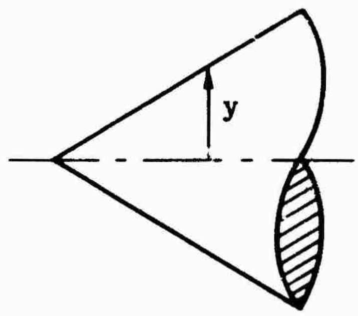
2) AXI2D (Option D-1)

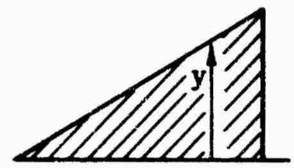
AXI2D calculates the flow conditions for arbitrary axisymmetric and two-dimensional geometries using the following procedure:

(a) With the geometry specified by the coordinates (ξ_i, y_i), the slopes and pressures are assumed to occur midway between the input geometry points.

$$\delta_{i+1/2} = \tan \frac{y_{i+1} - y_i}{\xi_{i+1} - \xi_i}$$

Where y is the local body coordinate:





and

$$x_{i+1/2} = x_{i-1/2} + \left[\left(y_{i+1/2} - y_{i-1/2}\right)^2 + \left(\xi_{i+1/2} - \xi_{i-1/2}\right)^2\right]$$

$$P_{i+1/2} = P_\infty \left[\frac{\gamma}{2} M_\infty^2 \frac{C_P}{\sin^2\delta} \sin^2\delta + 1\right]$$

where

$$\frac{C_P}{\sin^2\delta} = 0 \quad \text{for } \delta \leq 0$$

$$\frac{C_P}{\sin^2\delta} = 1.05 + \left[1.1025 + \frac{4}{M_\infty^2 \sin^2\delta}\right]^{1/2} - \frac{1.278 \sin^2\delta}{M_\infty^{.6}}$$

for $\delta > 0$

The end points x_I and x_f and the pressures at the end points are found by linear extrapolation.

(b) If P at either end point is less than P_∞, set the respective $P = P_\infty$. If P at either end point is greater than P_0, set the respective $P = P_0$.

(c) Initialize $x^* = x_I$ and calculate the flow conditions at each increment dx along the streamline

$$x_i^* = x_{i-1}^* + dx \qquad i \geq 2$$

Linear interpolation is performed to find $P/P_{SL}(x^*)$ at each x^* using the tables of P and x calculated previously.

If $P(x_I) \leq .528\ P_o$

$$u_e(x^*) = u_\infty \left[1 - \left(\frac{P}{P_\infty} - 1\right)\frac{1}{\gamma M_\infty^2}\right]^{1/2}$$

If $P(x_I) > .528\ P_o$ find the sonic location X^* where $P = .528\ P_o$ from interpolation of the x and P tables. Then

$$u_e(x^*) = \frac{a^* x}{X^*} \quad \text{when } x \leq X^*$$

$$u_e(x^*) = a^* \left\{6\left[1 - \left(\frac{P}{P_o}\right)^{2/7}\right]\right\}^{1/2} \quad \text{when } x > X^*$$

$$a^* = (H/3)^{1/2} \quad \text{Sonic velocity}$$

The remaining flow conditions are calculated from

$$\left.\begin{aligned} \frac{\Delta}{\Delta_i}(x^*) &= 1.0 \\ \frac{r}{r_i}(x^*) &= 1.0 \end{aligned}\right\} \quad \text{Two-dimensional body}$$

$$\left.\begin{aligned} \frac{\Delta}{\Delta_i}(x^*) &= y(x^*) \\ \frac{r}{r_i}(x^*) &= y(x^*) \end{aligned}\right\} \quad \text{Axisymmetric body}$$

$$i_e(x^*) = H - \frac{1}{2} u_e^2(x^*)$$

$$\left.\begin{aligned} \omega_e(x^*) &= f\left(P/P_{SL}(x^*),\ i_e(x^*)\right) \\ ZT_e(x^*) &= f\left(P/P_{SL}(x^*),\ i_e(x^*)\right) \end{aligned}\right\} \quad \text{[SOMEGA]}$$

(d) Step (c) is repeated until the flow properties have been calculated for all points on the streamline.

Input

H	H	total enthalpy, ft^2/sec^2
XSI	ξ	coordinate in free-stream flow direction, ft
Y	$y(\xi)$	geometry coordinate orthogonal to ξ, ft
XI	x_I	streamline coordinate at which calculations are to begin, ft
DX	dx	streamline coordinate increment, ft
POPSL	P_o/P_{SL}	stagnation pressure, atm
ACH	M_∞	free-stream Mach number
PINF	P_∞	free-stream pressure, lb/ft^2

Output

II		number of points calculated along streamline
XSTAR	x^*	streamline coordinate, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
UE	u_e	edge velocity, ft/sec
AIEE	$i_e(x^*)$	edge enthalpy, ft^2/sec^2
ZTEX	$(ZT)_e(x^*)$	edge compressibility-temperature product, °R
OMEGAE	$\omega_e(x^*)$	edge viscosity parameter, slug/ft-sec-°R
RORI	$(r/r_i)(x^*)$	divergence parameter due to body-shock layer geometry
DODI	$(\Delta/\Delta_i)(x^*)$	distance between streamlines, or total streamline divergence

3) BEQI

BEQI calculates the equivalent distance parameters at the initial streamline coordinate x_I. Following is the sequence of operations:

(a) Initialize $\quad \dfrac{b_{eq,L}}{x} = \dfrac{b_{eq,T}}{x} = 1.0$

(b) Divide the first streamline increment dx into 10 intervals, dx/10. Interpolate to find the flow properties at each interval.

(c) Interpolate for $\quad i_w = f(T_w, P/P_{SL})$ [IWALL]

$$\left.\begin{array}{l} ZT_S \\ \\ \omega_S \end{array}\right\} = f(H, P/P_{SL}) \quad \text{[SOMEGA]}$$

$$\left.\begin{array}{l} ZT_w \\ \\ \omega_w \end{array}\right\} = f(i_w, P/P_{SL}) \quad \text{[SOMEGA]}$$

(d) Calculate the reference terms

$$\omega_r,\ \rho_r \mu_r,\ ZT_r,\ i_r,\ \sigma_r \quad \text{[RORMUR]}$$

(e) Calculate

$$\frac{d(\Delta/\Delta_i)}{dx} \quad \text{[DERV]}$$

(f) Calculate $i_{e,SL}$ and EXPK from

$$N = \frac{x^* \, d(\Delta/\Delta_i)}{(\Delta/\Delta_i)\, dx}$$

If $N \le .05$; $.99 \le N \le 1.01$; EXPK = 0, $i_{e,SL} = i_e$

$.05 < N < .99$; EXPK = $-.194\, e^{-\frac{2}{3}N(N-1)}$, $\bar{\theta}_{SL} = (N-1)\,\theta_e$

$N > 1.01$; EXPK = $.194\, e^{-\frac{2}{3}(N-1)}$, $\bar{\theta}_{SL} = \left(\frac{N-1}{N}\right)\theta_e$

$$v_P = u_e \cos \bar{\theta}_{SL} ; \quad i_{e,SL} = H - \frac{v_P^2}{2}$$

(g) Obtain $\bar{E}_L$ and $\bar{E}_T$ [EBAR]

(h) Calculate the equivalent distance parameter

$$\left.\frac{b_{eq}}{x}\right|_{x*} = \frac{1}{x^* G(x^*)} \left\{ x_I G(x_I) \left(\frac{b_{eq}}{x}\right)_{x_I} + \int_{x_I}^{x*} G(x)dx \right.$$

where

$$G_L = \left(\rho_r \mu_r\right) u_e \left(r/r_i\right)^2 \left(f/f_i\right)^{2 \bar{E}_L x_I}$$

$$G_T = \left(\rho_r \mu_r\right) u_e \left(r/r_i\right)^{5/4} \left(f/f_i\right)^{(5/4)\bar{E}_T}$$

and

$$f/f_i = \frac{\Delta/\Delta_i}{r/r_i}$$

The integration is performed by applying Simpson's rule to the integrands G_L and G_T to find $b_{eq,L}/x$ and $b_{eq,T}/x$, respectively.

(i) b_{eq}/x is calculated for 10 intervals. The 10th b_{eq}/x is used as the new $(b_{eq}/x)_{x_I}$, the first interval is divided into 100 intervals, and the above calculations are repeated for the first 10 intervals.

(j) The calculations are repeated for 5 iterations (i) where

$$dx_i = \frac{dx_i}{10^i} \quad \text{and} \quad \left[\left(\frac{b_{eq}}{x}\right)_{x_I}\right]_i = \left[\left(\frac{b_{eq}}{x}\right)_{x_{II}}\right]_{i-1}$$

The final $\left[(b_{eq}/x)_{x_{II}}\right]_{i=5}$ is then used as the $(b_{eq}/x)_{x_I}$ for the general calculation (QTRANS).

Input

XSTAR	x^*	streamline coordinate at dx intervals, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
TW	$T_w(x^*)$	wall temperature, °R
UE	$u_e(x^*)$	edge velocity, ft/sec
THETAS	$\theta_e(x^*)$	streamline (edge) angle, degrees
OMEGAE	$\omega_e(x^*)$	edge viscosity parameter, slug/ft-sec-°R
DODI	$\Delta/\Delta_i(x^*)$	dimensionless distance between streamlines
RORI	$r/r_i(x^*)$	divergence parameter due to body-shock layer geometry
DX	dx	streamline coordinate increment, ft

Output

BEQXLI	$(b_{eq,L}/x)_{x_I}$	laminar equivalent distance parameter at x_I
BEQXTI	$(b_{eq,T}/x)_{x_I}$	turbulent equivalent distance parameter at x_I

4) CYLIND (Option D-3)

CYLIND calculates the coordinates ξ and η and the two-dimensional parameters $\theta_e(\xi,\eta)$, $P/P_o(\xi,\eta)$ and $T_w(\xi,\eta)$ for an infinite swept cylinder. The properties are calculated as a function of η at $\xi = 0$ and are assumed to be constant in the ξ direction.

Pressure: $$\frac{P(\eta)}{P_o} = \left[\frac{P(\theta)}{P_{T_2}}\right] \times \left[\frac{P_{T_2}}{P_o}\right]$$

where

$$\theta = 57.3 \frac{\eta}{R_{CYL}}$$

$P/P_{T_2}(\theta)$ is found by linear interpolation on a built-in table (Figure 1).

$$P_{T_2} = P_\infty \left(1.2\,M_N\right)^{3.5} \left(\frac{6}{7M_N^2 - 1}\right)^{2.5} \qquad \text{Stagnation Line Pressure}$$

$$M_N = M_\infty \cos \Lambda$$

Wall temperature is found by linear interpolation on the input array.

Streamline angle:

$$\theta_e(\eta) = \tan^{-1} \frac{v(\eta)}{u_\infty \sin \Lambda}$$

where $v(\eta)$ is the tangential velocity calculated in subroutine FLOW.

Then

$$P/P_o(\xi, \eta) = P/P_o(\eta) \qquad \text{for all } \xi$$

$$T_w(\xi, \eta) = T_w(\eta) \qquad \text{for all } \xi$$

$$\theta_e(\xi, \eta) = \theta_e(\eta) \qquad \text{for all } \xi$$

$$(u_e)_{x_I} = u_\infty \sin \Lambda$$

<u>Input</u>

RADIUS	R_{CYL}	cylinder radius, ft
SWEEP	Λ	sweep angle, degrees
ACH	M_∞	free-stream Mach number
VEL	u_∞	free-stream velocity, ft/sec
ETAF	η_f	final η coordinate for which streamline is to be calculated, ft
XSIF	ξ_f	final ξ coordinate for which streamline is to be calculated, ft
TW	T_w	wall temperature, °R

PINF	P_∞	free stream pressure, lb/ft^2
POPSL	P_o/P_{SL}	stagnation pressure, atm

<u>Output</u>

XSI	ξ	coordinate along axis of cylinder, ft
ETA	η	coordinate tangential to cylinder, ft
M		number of ξ values
N		number of η values
ETAMAX	$\eta_{max}(\xi)$	maximum η value, ft
TH8MX	$\theta_{max}(\xi)$	streamline angle at η_{max}, degrees
PRESSR	$P/P_o(\xi, \eta)$	pressure ratio array
TWALL	$T_w(\xi, \eta)$	wall temperature array, °R
THETAE	$\theta_e(\xi, \eta)$	streamline angle array, degrees

5) DBTP

DBTP is a linear double interpolation routine. Search time is reduced by starting the current search at the search result for the previous entry to this routine. DBTP uses TBLP to perform the interpolation.

<u>Input</u>

XX	independent variable table
YY	independent variable table
ZZ	dependent variable table
X	independent variable
Y	independent variable
NX	number of values in X table
NY	number of values in Y table

NXS location in X table at which search is begun

NYS location in Y table at which search is begun

Output

Z dependent variable

6) DELTA (Option D-4)

DELTA calculates the body geometry, the streamline angles, and the two-dimensional pressure array $P/P_0(\xi, \eta)$ for a sharp delta wing. The following procedure is used:

(a) If $R \leq 0$ a sharp delta wing case is indicated and $\delta = \alpha$ (degrees) for all ξ.

(b) The pressure is calculated from

$$P(\xi, \eta) = \frac{\gamma}{2} P_\infty M_\infty^2 \left(\frac{C_P}{\sin^2 \delta} \right) \sin^2 \delta + P_\infty$$

where

$$\frac{C_P}{\sin^2 \delta} = 1.05 + \left[1.1025 + \frac{4}{M_\infty^2 \sin^2 \delta} \right]^{1/2} - \frac{1.278 \sin^2 \delta}{M_\infty^{.6}}$$

(c) The initial velocity is calculated from

$$(u_e)_{x_I} = u_\infty \left[1 - \delta^2/5600 \right]$$

(d) The equations used to calculate $\theta_e(\xi, \eta)$ are:

$$\mathcal{L} = \left\{ \left[\delta + 1.22 \left(\frac{\rho_\infty}{\rho_2} \right) \left(\frac{\tan \delta}{\tan \Lambda} \right)^{.566} \right]^2 + \left[\sin^{-1} \frac{1}{M_\infty} \right]^2 \right\}^{1/2}$$

$$M_N = M_\infty \sin \mathcal{L}$$

$$\bar{\Phi}^{**} = \frac{1}{90 - \Lambda} \tan^{-1} \left\{ \frac{1}{6} \left[1 + \frac{5}{M_\infty^2} \left(\frac{M_N^2 - 1}{M_N^2} \right) \right] \frac{1}{(u/u_\infty)^2} + \frac{1}{6} \right\}^{1/2}$$

$$\gamma = \frac{5(\bar{\Phi}^{**})^4}{5(\bar{\Phi}^{**})^4 + 1}$$

$$\theta^* = (90 - \Lambda)(1 + \bar{\Phi}^{**})\gamma$$

$\bar{N}_{CL}$ is found by linear interpolation on a built-in table (Figure 3).

If $\bar{\Phi}^{**} \le 1.2$ $\quad \bar{N}_{CL} = f(\bar{\Phi}^{**})$

If $\bar{\Phi}^{**} > 1.2$ $\quad \bar{N}_{CL} = f(\Lambda, \bar{\Phi}^{**})$

$$N_{CL} = \frac{1 + \bar{\Phi}^{**}}{\frac{M_N^2 + 1}{M_N^2}} \bar{N}_{CL}$$

$$\eta_{max} = \xi \tan(90 - \Lambda)$$

$$\theta = (90 - \Lambda) N_{CL} \bar{R} + \left[\theta^* - (90 - \Lambda) N_{CL}\right] \bar{R}^9$$

where

$$\bar{R} = \frac{\tan(\eta/\xi)}{\tan(90 - \Lambda)}$$

Then

$$\theta_{max} = \theta^*$$

$$\xi_i = \xi_{i-1} + \Delta\xi$$

$$\eta_i = \eta_{i-1} + \Delta\eta$$

(e) The delta wing option D-4 assumes zero streamline divergence due to body geometry

$$\frac{r}{r_i} = 1.0$$

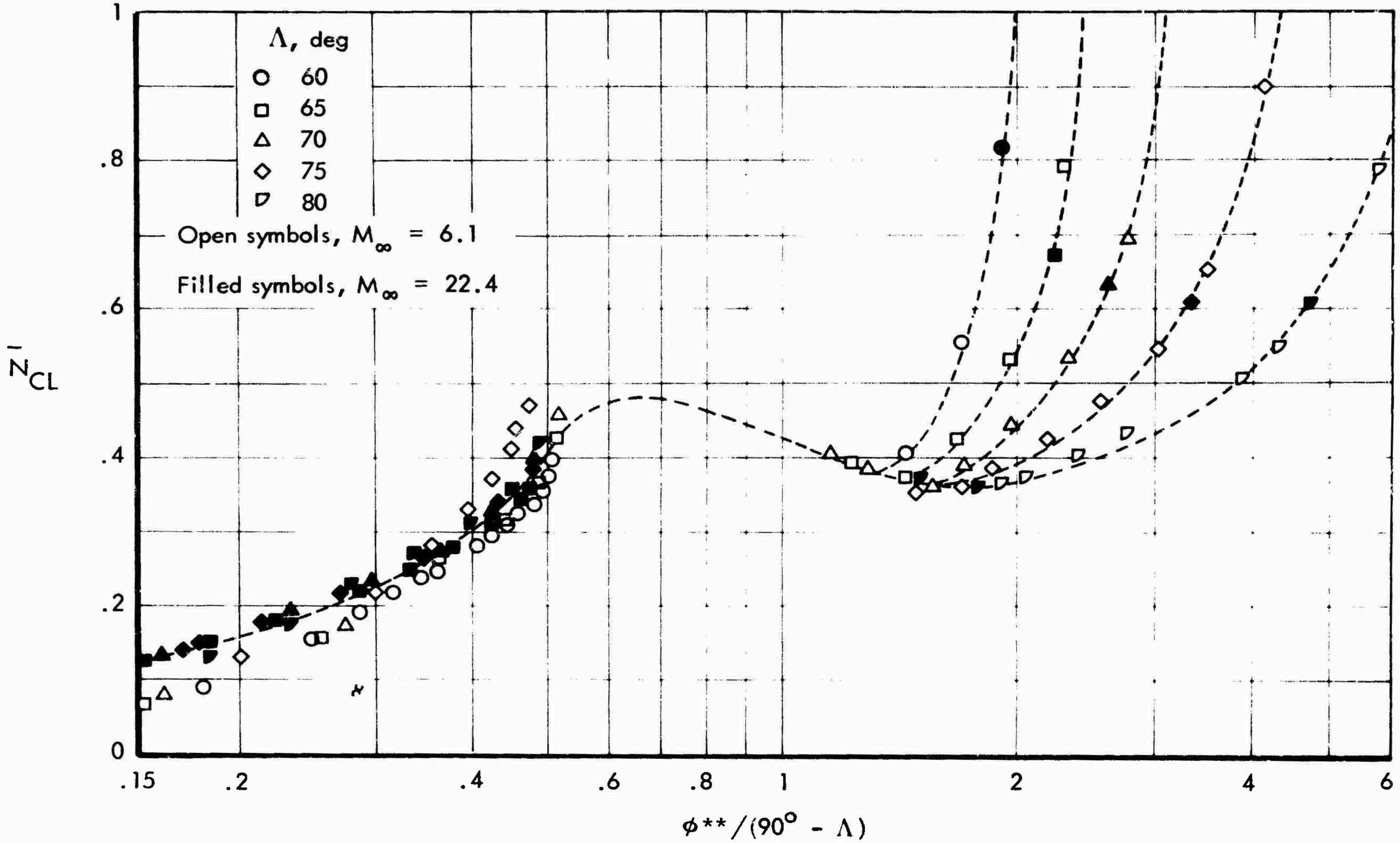


Figure 3: STREAMLINE CORRELATION FOR SHARP DELTA WINGS

Input

ACH	M_∞	free stream Mach number
PINF	P_∞	free stream pressure, lb/ft^2
POPSL	P_o/P_{SL}	stagnation pressure, atm
VEL	u_∞	free stream velocity, ft/sec
ALPHA	α	angle of attack, degrees
SWEEP	Λ	sweep angle, degrees
RADIUS	R	leading edge radius, ft
XSII	ξ_I	initial streamline ξ coordinate, ft
ETAI	η_I	initial streamline η coordinate, ft
XSIF	ξ_f	final streamline ξ coordinate, ft
ETAF	η_f	final streamline η coordinate, ft
DELXSI	$\Delta\xi$	increment used to calculate ξ coordinate, ft
DELETA	$\Delta\eta$	increment used to calculate η coordinate, ft

Output

M		number of ξ values
N		number of η values
XSI	ξ	geometry coordinate, ft
ETA	η	geometry coordinate, ft
ETAMAX	η_{max}	leading edge coordinate, ft
TH8MX	θ_{max}	streamline angle at leading edge, degrees
THETAE	θ_e	streamline angle, degrees
PRESSR	P/P_o	pressure ratio
URATIO	u/u_∞	velocity ratio

7) DERV (V, W, X, Y, Z, DBYDX, DX, ITYPE)

DERV calculates the derivative of a tabulated function f by Langrangian five-point formulas. Given a sequence of points x_{-2}, x_{-1}, x_o, x_1, x_2 separated by a common increment h, the corresponding derivative of f at each point is given by:

$$f'_{-2} = \frac{1}{h}(-f_{-2} + f_{-1}) \; ; \; f'_{-1} = \frac{1}{2h}(-f_{-2} + f_o)$$

$$f'_o = \frac{1}{2h}(-f_{-1} + f_1) \; ; \; f'_1 = \frac{1}{2h}(-f_o + f_2)$$

$$f'_2 = \frac{1}{h}(-f_1 + f_2)$$

Input

V	f_{-2}	functional values
W	f_{-1}	
X	f_o	
Y	f_1	
Z	f_2	
ITYPE		determines which derivative formula is to be used
DX	h	function increment

Output

DBYDX	f'	function derivative

8) DIVERG

DIVERG calculates P/P_o, T_w, θ_e, r/r_i and Δ/Δ_i along a given streamline by the following method:

(a) Store the original streamline (1) data and calculate an auxiliary streamline (2) such that

$$\eta_{F_2} = \eta_{F_1} \pm \Delta\eta_{max}$$

(b) Calculate the following at each dx location on the streamline of interest

$\theta_e(x^*)$	linear interpolation on $\theta_e(x)$
$T_w(x^*)$	double linear interpolation on $T_w(\xi, \eta)$
$P/P_o(x^*)$	double linear interpolation on $P/P_o(\xi, \eta)$

$$\frac{r}{r_i}(x^*) = 1.0$$

$$\frac{\Delta}{\Delta_i}(x^*) = \frac{\eta_1(x^*) - \eta_2(x^*)}{\eta_{i1}(x^*) - \eta_{i2}(x^*)} \; \frac{\cos\theta_{e_1}(x^*)}{\cos\theta_{e_{i1}}(x^*)}$$

Where the subscripts 1, 2, and i indicate the original and auxiliary streamlines and the point at which calculations are begun, respectively. If the streamlines cross at any point, the above calculations are begun a distance dx downstream from the intersection point.

Input

XI	x_I	initial streamline distance, ft
DX	dx	streamline coordinate increment, ft
ETAF	η_f	final streamline coordinate, ft
XSIF	ξ_f	final streamline coordinate, ft
M		number of ξ input values
N		number of η input values
XSI	ξ	streamline coordinate array, ft
ETA	η	streamline coordinate array, ft
PRESSR	$P/P_o(\xi, \eta)$	pressure ratio array
TWALL	$T_w(\xi, \eta)$	wall temperature array, °R
XSIX	$\xi(x)$	streamline coordinate, ft
ETAX	$\eta(x)$	streamline coordinate, ft

X	x	streamline coordinate, ft
ETAMAX	η_{max}	η coordinate for geometry leading edge, ft
THETAX	$\theta_e(x)$	streamline angle, degrees

<u>Output</u>

XSTAR	x^*	streamline coordinate, ft
PRESRX	$P/P_o(x^*)$	pressure ratio along streamline
THETAS	$\theta_e(x^*)$	streamline angle, degrees
DODI	$(\Delta/\Delta_i)(x^*)$	dimensionless distance between streamlines
RORI	$(r/r_i)(x^*)$	divergence parameter
TW	T_w	wall temperature, °R
XF	x_f	final streamline x-coordinate, ft
II		number of points on streamline

9) DSIRCH (Z, X, Y, ZA, XA, YA, NZ, NX, NY, IX, IY)

DSIRCH uses DTAB to perform double interpolation on a table of the form

$$Z = f(x, y)$$

<u>Input</u>

X, Y	independent variables
XA	table of X values
YA	table of Y values
ZA	table of Z values
NX	number of values in X table
NY	number of values in Y table
NZ	number of values in Z table

IX order of interpolation in X direction

IY order of interpolation in Y direction

<u>Output</u>

Z dependent variable

10) EBAR (AYEW, AYESL, H, SIGR, EXPK, POPSLX, OMEGE, OMEGS, GAMC, GAMO, EBARL, EBART)

EBAR calculates the crossflow pressure gradient parameters $\bar{E}_L$ and $\bar{E}_T$ and the pressure gradient correlation parameters Γ_c and Γ_o using the following method:

(a) Interpolate to find

$$ZT_{e,SL} = f(P/P_{SL}, i_{e,SL}) \quad \text{[SOMEGA]}$$

(b) Calculate the local pressure in lb/ft^2 and the $\rho\mu$ terms

$$P_r = P_{SL}\left(\frac{P}{P_{SL}}\right)$$

$$\rho_T\mu_T = \frac{P_r}{R}\,\omega_s$$

$$\rho_e\mu_e = \frac{P_r}{R}\,\omega_e$$

(c) Calculate the mean enthalpies

$$i_{avg} = \frac{1}{2}(i_w - i_{e,SL})$$

$$i_{m,c} = i_{avg} + .2\,(H - i_{e,SL})\,\sigma_r\left(\frac{\rho_T\mu_T}{\rho_e\mu_e}\right)^{1/2}$$

(d) Interpolate to find

$$ZT_{m,c} = f(P/P_{SL}, i_{m,c}) \quad \text{[SOMEGA]}$$

$$ZT_{m,o} = f(P/P_{SL}, i_{avg}) \quad \text{[SOMEGA]}$$

(e) Compute:

$$F_{\Sigma_c} = \frac{(\Sigma_c - .294)\,\sigma_r^{.355}}{.4018}$$

$$F_{\Sigma_o} = \frac{(\Sigma_o - .294)\,\sigma_r^{.355}}{.4018}$$

and

$$F_{K_c} = \left(2\Sigma_c\right)^{EXPK}$$

$$F_{K_o} = \left(2\Sigma_o\right)^{EXPK}$$

where

$$\Sigma_c = \frac{ZT_{m,c}}{ZT_{e,SL}}$$

$$\Sigma_o = \frac{ZT_{m,o}}{ZT_{e,SL}}$$

(f) The pressure gradient correlation parameters are calculated from:

$$\Gamma_c = .71764\, F_{K_c}\left(\sqrt{1 + F_{\Sigma_c}} - 1\right)$$

$$\Gamma_o = .71764\, F_{K_o}\left(\sqrt{1 + F_{\Sigma_o}} - 1\right)$$

(g) Compute the crossflow pressure gradient parameters:

$$\overline{E}_L = 1 + \Gamma_c$$

$$\overline{E}_T = (1 + .76519\,\Gamma_o)\left(\frac{1 + \Gamma_c}{1 + \Gamma_o}\right)^4$$

Input

AYEW	i_w	wall enthalpy, ft^2/sec^2
AYESL	$i_{e, SL}$	edge enthalpy at flow line of symmetry, ft^2/sec^2
H	H	total enthalpy, ft^2/sec^2
SIGR	σ_r	Prandtl number
POPSLX	P/P_{SL}	local pressure along streamline, atm
OMEGE	ω_e	edge viscosity parameter, slug/ft-sec-°R
OMEGS	ω_s	stagnation viscosity parameter, slug/ft-sec-°R

Output

GAMC	Γ_c	pressure gradient correlation parameter
GAMO	Γ_o	pressure gradient correlation parameter
EBARL	$\overline{E}_L$	laminar crossflow pressure gradient parameter
EBART	$\overline{E}_T$	turbulent crossflow pressure gradient parameter

11) FIND (ZA, Z, NZ, NZS)

FIND looks up **Z** in an array ZA, dimensioned NZ, and stores the location of Z relative to ZA(1) in NZS. If Z is not in the range of ZA, the message "VARIABLE NOT IN RANGE OF TABLE" is printed, followed by the input values of Z, ZA(1), and ZA(NZ), respectively. NZS is then set to 1 if Z < ZA(1) and to NZ if Z > ZA(NZ). Upon subsequent entries to FIND the search is begun at ZA(NZS) where NZS is now the location of Z from the previous entry to FIND. For a normal exit NZS = i where $ZA_i \le Z < ZA_{i+1}$.

Input

ZA	array to be searched
Z	variable for which location is desired

NZ	number of values in ZA
NZS	location of Z in ZA from previous call to FIND

Output

NZS	location of Z in ZA

12) FLOW (Program B)

FLOW calculates the external flow parameters at the boundary layer edge as a function of streamline distance given the pressure and the stagnation conditions.

At the starting point on the streamline, edge enthalpy is obtained from:

$$\left(i_e\right)_{x_I} = H - \frac{\left(u_e\right)^2_{x_I}}{2}$$

For the special case of the swept cylinder tangential velocity calculation

$$\left(i_e\right)_{x_I} = H - \frac{\left(v^2\right)_{x_I}}{2} - \frac{\left(u_\infty \sin \Lambda\right)^2}{2}$$

Then

$$ZT_e = f(i_e, P/P_{SL}) \qquad [SOMEGA]$$

$$\omega_e = f(i_e, P/P_{SL}) \qquad [SOMEGA]$$

Let

$$u = \frac{1}{2}\left(\frac{u_e}{u_\infty}\right)^2$$

and define

$$\left|u_{x^*}\right|_K \equiv \left\{\frac{1}{2}\left(\frac{u_e}{u_\infty}\right)^2\right\}_{x^*, K}$$

as the Kth iteration for u at the point x*.

Similarly define $\left|(ZT_e)_{x^*}\right|_K$ as the Kth iteration for ZT_e at x^*.

The iterative equation to be satisfied at each point x^* on the streamline is

$$\left|u_{x^*}\right|_K = u_{x^*-dx} - \frac{dxR}{u_\infty^2}\left|\left(ZT_e\right)_{x^*}\right|_K \frac{\left[\frac{d}{dx}\left(\frac{P}{P_o}\right)\right]_{avg}}{\left(P/P_o\right)_{avg}}$$

where

$$\frac{d(P/P_o)}{dx} = \frac{1}{2}\left\{\frac{d}{dx}(P/P_o)\Big|_{x^*} + \frac{d}{dx}(P/P_o)\Big|_{x^*-dx}\right\}$$

and

$$\left(P/P_o\right)_{avg} = \frac{1}{2}\left\{\frac{P}{P_o}(x^*) + \frac{P}{P_o}(x^* - dx)\right\}$$

The derivatives are calculated numerically using subroutine DERV.

For each iteration:

$$\left|\left(i_e\right)_{x^*}\right|_K = H - u_\infty^2\left|u_{x^*}\right|_K$$

For the swept cylinder tangential velocity

$$\left|\left(i_e\right)_{x^*}\right|_K = H - u_\infty^2\left|u_{x^*}\right|_K - \frac{1}{2}\left(u_\infty \sin\Lambda\right)^2$$

Then

$$\left.\begin{array}{l}\left|\left(ZT_e\right)_{x^*}\right|_K = \\ \left|\left(\omega_e\right)_{x^*}\right|_K = \end{array}\right\} f\left[\left|\left(i_e\right)_{x^*}\right|_K, \frac{P}{P_{SL}}(x^*)\right] \quad \text{[SOMEGA]}$$

This value of ZT_e is then used in the above iterative equation to yield a new u_{x*}. These calculations are repeated at every $x*$ until

$$\left| \frac{\left|u_{x*}\right|_K - \left|u_{x*}\right|_{K-1}}{\left|u_{x*}\right|_K} \right| < \epsilon$$

where $\epsilon = 10^{-6}$. A maximum of ten iterations (K = 10) is fixed by the program. If the relative error criteria has not been satisfied after ten iterations at any point $x*$, an appropriate error comment will be printed. If convergence at $x*$ has been established in the Ith step, then set $u_{x*} = \{u_{x*}\}_I$ and

$$\left(u_e\right)_{x*} = u_\infty\left(2u_{x*}\right)^{1/2}$$

$$\left(i_e\right)_{x*} = \left|\left(i_e\right)_{x*}\right|_I$$

$$\left(ZT_e\right)_{x*} = \left|\left(ZT_e\right)_{x*}\right|_I$$

$$\left(\omega_e\right)_{x*} = \left|\left(\omega_e\right)_{x*}\right|_I$$

A special case results when $\left(u_e\right)_{x_I} = 0$. This corresponds to a stagnation point where $x_I = 0$ and $\left|d/dx(P/P_o)\right|_{x_I} = 0$. The iterative equation for u_e becomes $u_e(x*) = u_e(x* - dx)$ or $du_e/dx = 0$. Because this result is incorrect, the first step is calculated using the following:

The velocity gradient at the stagnation point x_I is calculated from:

$$\left[\frac{d}{dx}\left(\frac{u_e}{u_\infty}\right)\right]_{x_I} = \frac{1}{u_\infty}\left\{-R\left(ZT_e\right)_{x_I} \frac{d^2}{dx^2}\left(\frac{P}{P_o}\right)_{x_I}\right\}^{1/2}$$

where the pressure derivative is obtained from the numerical derivative equation

$$\frac{d^2}{dx^2}\left(\frac{P}{P_o}\right) = \frac{\frac{P}{P_o}(\xi_1, \eta) - 2\frac{P}{P_o}(\xi_2, \eta) + \frac{P}{P_o}(\xi_3, \eta)}{\Delta\xi^2}$$

The use of this equation requires that the first three ξ input points be equidistant. Then

$$\left(u_e\right)_{x_{I+dx}} = u_\infty \left[\frac{d}{dx}\left(\frac{u_e}{u_\infty}\right)\right]_{x_I} dx$$

$$\left(i_e\right)_{x_{I+dx}} = H - \frac{1}{2}\left(u_e^{\,2}\right)_{x_{I+dx}}$$

$$\left.\begin{array}{l}\left(ZT_e\right)_{x_{I+dx}} \\ \left(\omega_e\right)_{x_{I+dx}}\end{array}\right\} = f\left[\left(i_e\right)_{x_{I+dx}}, \left(\frac{P}{P_{SL}}\right)_{x_{I+dx}}\right] \quad \text{[SOMEGA]}$$

The calculations continue, using the previously described procedure where the starting point is now $x_I + dx$.

Input

PRESRX	$P/P_o(x^*)$	ratio of local to stagnation pressure along streamline
II		number of values in pressure array
URATIO	$(u_e/u_\infty)_{x_I}$	velocity ratio at x_I
H	H	total enthalpy, ft^2/sec^2
VEL	u_∞	free stream velocity, ft/sec
DX	dx	streamline coordinate increment, ft
POPSL	P_o/P_{SL}	stagnation pressure, atm

Output

		The following outputs are a function of streamline distance:
UE	u_e	edge velocity, ft/sec
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm

AIEE	i_e	edge enthalpy, ft^2/sec^2
ZTEX	ZT_e	edge compressibility-temperature product, °R
OMEGAE	ω_e	edge viscosity temperature ratio, slug/ft-sec-°R
XSTAR	x^*	streamline coordinate, ft

13) HEMI (Option D-2)

HEMI calculates the streamline and the pressure and divergence parameters along the streamline for a hemisphere geometry.

(a) Set $x_I^* = x_I$

(b) Calculate x^*, $P/P_o(x^*)$, $r/r_i(x^*)$, and $\Delta/\Delta_i(x^*)$ for all dx increments along the streamline until $\theta = 90°$

$$\theta = \frac{57.3\, x^*}{R_{HEMI}}$$

Interpolate to find $P/P_o(x^*) = f(\theta)$

$$r/r_i(x^*) = \sin\left(\frac{x^*}{R_{HEMI}}\right)$$

$$\Delta/\Delta_i(x^*) = \sin\left(\frac{x^*}{R_{HEMI}}\right)$$

(c) URATIO = $u_{e_{x_I}}/u_\infty$

<u>Input</u>

RHEM	R_{HEMI}	hemisphere radius, ft
XI	x_I	x-value at which calculations are to begin, ft
DX	dx	streamline increment, ft

<u>Output</u>

XSTAR	x^*	streamline coordinate, ft
PRESRX	$P/P_o(x^*)$	ratio of local to stagnation pressure along streamline

RORI	$r/r_i(x^*)$	divergence parameter due to body-shock layer geometry
DODI	$\Delta/\Delta_i(x^*)$	distance between streamlines, total streamline divergence

14) INTIAL (L1, NTYPE)

INTIAL calculates the free stream conditions for wind tunnel and flight type inputs.

(a) Wind Tunnel Class I

$$u_\infty = 49\ M_\infty \sqrt{T_\infty}$$

$$H = 6000\ T_\infty + \frac{u_\infty^2}{2}$$

$$q_\infty = 0.7\ P_\infty M_\infty^2$$

$$\rho_\infty = P_\infty/(1716\ T_\infty)$$

(b) Wind Tunnel Class II

$$i_\infty = H - \frac{u_\infty^2}{2}$$

$$a_\infty = f(i_\infty) \quad \text{when } 5.93 \times 10^6 \le i_\infty \le 20.3 \times 10^6\ \text{ft}^2/\text{sec}^2 \quad \text{[SONENT]}$$

$$a_\infty = .6325 \sqrt{i_\infty} \quad \text{when } i_\infty < 5.93 \times 10^6\ \text{ft}^2/\text{sec}^2$$

$$M_\infty = \frac{u_\infty}{a_\infty}$$

$$T_\infty = \left(\frac{u_\infty}{49\ M_\infty}\right)^2$$

$$\rho_\infty = P_\infty/(1716\ T_\infty)$$

P_o/P_∞ from [STACON]

$$P_\infty = \left(\frac{P_o}{P_{SL}}\right) P_{SL} \left(\frac{P_\infty}{P_o}\right)$$

$$q_\infty = \frac{P_\infty u_\infty^2}{2RT_\infty}$$

(c) Flight

$$\left.\begin{aligned} a_\infty &= f\,(\text{altitude}) \\ P_\infty &= f\,(\text{altitude}) \\ T_\infty &= f\,(\text{altitude}) \end{aligned}\right\} \quad \text{[ATMOS]}$$

$$H = 6000\ T_\infty + \frac{u_\infty^2}{2}$$

$$M_\infty = \frac{u_\infty}{a_\infty}$$

$$q_\infty = 0.7\ P_\infty M_\infty^2$$

For wind tunnel I and flight

$P_o/P_\infty = f\,(M_\infty,\ H,\ P_\infty,\ u_\infty)$ [STACON]

$$P_o/P_{SL} = \left(\frac{P_o}{P_\infty}\right) \frac{P_\infty}{P_{SL}}$$

<u>Input</u>

Wind Tunnel Class I, NTYPE = WT1

ACH	M_∞	free stream Mach number
TEMP	T_∞	free stream temperature, °R
PINF	P_∞	free stream pressure, lb/ft^2

Wind Tunnel Class II, NTYPE = WT2

VEL	u_∞	free stream velocity, ft/sec
H	H	total enthalpy, ft^2/sec^2
POPSL	P_o/P_{SL}	stagnation pressure, atm
PINF	P_∞	free stream pressure, lb/ft^2

Flight, NTYPE = FLIGHT

ALT		flight altitude, ft
VEL	u_∞	free stream velocity, ft/sec

Output

ACH	M_∞	free stream Mach number
H	H	total enthalpy, ft^2/sec^2
PINF	P_∞	free stream pressure, lb/ft^2
POPINF	P_o/P_∞	stagnation to free stream pressure ratio
POPSL	P_o/P_{SL}	stagnation pressure, atm
QINF	q_∞	free stream dynamic pressure, lb/ft^2
SONIC	a_∞	speed of sound, ft/sec
VEL	u_∞	free stream velocity, ft/sec

15) IWALL (TW, POPSLX, AYEW)

IWALL calculates the enthalpy at the wall as a function of the wall temperature and the local pressure.

(a) Set $T_{W_K} = T_W/1.8$ (conversion to °K)

(b) Calculate wall enthalpy

$$i_w = .432\, T_{w_K} \quad \text{when} \quad T_{w_K} < 300°K$$

$$i_w = .432\, T_{w_K} + 3.82 \times 10^{-5} \left(T_{w_K} - 300\right)^2 \quad \text{when} \quad 300 < T_{w_K} < 1800°K$$

$$i_w = f\left(\log_{10} P/P_{SL},\ T_{w_K}\right); \ \left|\text{TABLE 6}\right| \quad \text{when} \quad T_{w_K} > 1800°K$$

(c) Convert i_w to ft^2/sec^2

$$i_w\left(\frac{ft}{sec^2}\right) = 25031\ i_w\left(\frac{Btu}{lb}\right)$$

Input

TW	T_w	wall temperature, °R
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm

Output

AYEW	i_w	wall enthalpy, ft^2/sec^2

16) JAYELL (AYEW, H, AYEE, ZTE, ZTS, SIGR, BETAS, POPSLX, XJL)

JAYELL calculates the laminar pressure gradient effect correlation parameter J_L from the following:

(a) Calculate the mean enthalpy

$$i_{m,s} = \frac{1}{2}(i_w + H)$$

(b) Interpolate to find:

$$ZT_{m,s} = f(P/P_{SL},\ i_{m,s}) \quad [\text{SOMEGA}]$$

(c) Calculate

$$F_{\Sigma_s} = \frac{(\Sigma_s - .294)}{.4018}\left(\frac{i_{aw}}{H}\right)\sigma_r^{.355}$$

where

$$\Sigma_s = \frac{ZT_{m,s}}{ZT_s}$$

$$i_{aw} = H\left[\frac{i_e}{H} + \sigma_r^{1/2}\left(1 - \frac{i_e}{H}\right)\right]$$

(d) If $\beta_s < 0$ set $j = -1$

If $\beta_s \geq 0$ set $j = 1$

See subroutine QTRANS for β_s calculation.

(e) Calculate the streamwise pressure gradient function:

$$F_{\beta_s} = \frac{(1 + 2\,CPES)\,|\beta_s|}{2\,CPES + \left(\frac{j+1}{2}\right)\beta_s}$$

where

$$CPES = \frac{ZT_e/ZT_s}{i_e/H}$$

(f) Calculate J_L

$$J_L = \left[1 + .7176\left(\sqrt{1 + F_{\beta_s} F_{\Sigma_s}} - 1\right)\right]^j$$

<u>Input</u>

AYEW	i_w	wall enthalpy, ft^2/sec^2
H	H	total enthalpy, ft^2/sec^2
AYEE	i_e	edge enthalpy, ft^2/sec^2
ZTE	ZT_e	edge compressibility-temperature product, °R
ZTS	ZT_s	stagnation compressibility-temperature product, °R

SIGR	σ_r	Prandtl number
BETAS	β_S	streamwise pressure gradient parameter
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm

Output

XJL	J_L	laminar pressure gradient effect correlation parameter

17) MUZERO (OMEGR, ZTR, AYER, AYEE, H, POPSLX. XMUO, XL)

MUZERO computes the reference stagnation viscosity μ_0 and the diffusion effect parameter $\mathcal{L}$.

(a) Interpolate to find:

$$\frac{i_D}{i_e} = f(\log_{10} P/P_{SL},\ i_e) \qquad \text{[TABLE 7]}$$

If

$$i_D/i_e < 0, \quad \text{set } i_D/i_e = 0$$

(b) Calculate μ_0 and $\mathcal{L}$

$$\mu_0 = \omega_r(ZT_r)\left(\frac{H}{i_r}\right)^{3/2}\left[\frac{ZT_r + 200}{\left(\frac{H}{i_r}\right) ZT_r + 200}\right]$$

$$\mathcal{L} = 1 + .19\left(\frac{i_D}{i_e}\right)\frac{i_e}{H}$$

Input

OMEGR	ω_r	reference viscosity-temperature ratio, slug/ft-sec-°R
ZTR	ZT_r	reference compressibility-temperature product, °R
AYER	i_r	reference enthalpy, ft^2/sec^2

AYEE	i_e	edge enthalpy, ft^2/sec^2
H	H	total enthalpy, ft^2/sec^2
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
<u>Output</u>		
XMVO	μ_o	reference stagnation viscosity, slug/ft-sec
XL	$\mathcal{L}$	parameter indicating diffusion effect on heat transfer

18) QTRANS (Program A)

QTRANS calculates heat transfer data for any body shape given the external flow properties, gas properties at the wall, gas properties at the stagnation conditions, and the streamline divergence parameters. The sequence of calculations is as follows (names in brackets are subroutines):

(a) Set up x-printout array

(b) Initialize $\beta_s = 0.5$

(c) Interpolate for

$$i_w = f(T_w, P/P_{SL}) \qquad [IWALL]$$

(d) Interpolate to find

$$\left.\begin{matrix} ZT_s \\ \omega_s \end{matrix}\right\} = f(H, P/P_{SL}) \qquad [SOMEGA]$$

$$\left.\begin{matrix} ZT_w \\ \omega_w \end{matrix}\right\} = f(i_w, P/P_{SL}) \qquad [SOMEGA]$$

(e) Calculate

$$\omega_r, \ \rho_r\mu_r, \ ZT_r, \ i_r, \ \sigma_r \qquad [RORMUR]$$

(f) Calculate

$$\frac{du_e}{dx}, \quad \frac{d(\Delta/\Delta_i)}{dx} \qquad \text{[DERV]}$$

(g) Calculate $i_{e,SL}$ and EXPK from

$$N = \frac{x^*}{\frac{\Delta}{\Delta_i}} \frac{d(\Delta/\Delta_i)}{dx}$$

If $N \leq .05$; $.99 \leq N \leq 1.01$; EXPK = 0, $i_{e,SL} = i_e$

$.05 < N < .99$; EXPK = $-.194e^{-\frac{2}{3}N(N-1)}$, $\bar{\theta}_{SL} = (N-1)\,\theta_e$

$N > 1.01$; EXPK = $.194\,e^{-\frac{2}{3}(N-1)}$, $\bar{\theta}_{SL} = \left(\frac{N-1}{N}\right)\theta_e$

(h) Obtain $\bar{E}_L$ and $\bar{E}_T$ [EBAR]

(i) Calculate the equivalent distance parameters defined by:

$$\frac{b_{eq}}{x} = \frac{1}{x^*\,G(x^*)}\left\{x_i\,G(x_i)\left(\frac{b_{eq}}{x}\right)_{x_i} + \int_{x_i}^{x^*} G(x)dx\right\}$$

$$= \frac{1}{x^*\,G(x^*)}\left\{x_i\,G(x_i)\left(\frac{b_{eq}}{x}\right)_{x_i} + \int_{x_i}^{x^*-2dx} G(x)dx + \int_{x^*-2dx}^{x^*} G(x)dx\right\}$$

where the integrand G is calculated from

$$G_L = (\rho_r\mu_r)u_e\left(\frac{r}{r_i}\right)^2\left(\frac{f}{f_i}\right)^{2\bar{E}_L}$$

$$G_T = (\rho_r\mu_r)u_e\left(\frac{r}{r_i}\right)^{5/4}\left(\frac{f}{f_i}\right)^{(5/4)\bar{E}_T}$$

and

$$\frac{f}{f_i} = \frac{\Delta/\Delta_i}{r/r_i}$$

Thus, for $x^* = x_i + (2K)dx$ (INTEGER $x \geq 1$) the integration is performed by applying Simpson's rule using G_L or G_T as the integrand to find $(b_{eq}/x)_L$ and $(b_{eq}/x)_T$, respectively.

$$\left(\frac{b_{eq}}{x}\right)_{x^*} = \frac{1}{x^* G(x^*)} \left\{ x_i G(x_i) \left(\frac{b_{eq}}{x}\right)_{x_i} + \int_{x_i}^{x^*-2dx} G(x)dx + \frac{dx}{3}\left[G(x^* - 2dx) + 4\,G(x^* - dx) + G(x^*)\right]\right\}$$

where

$$\int_{x_i}^{x^*-2dx} G(x)dx = \frac{dx}{3}\sum_{j=1}^{K-1}\left\{ G\left[x_i + (2j-2)dx\right] + 4\left[x_i + (2j-1)dx\right] + G(x_i + 2jdx)\right\}$$

This last sum is the result of successive application of the three point integration as the calculations move along the streamline. When $x^* = x_i + (2K+1)dx$ an extra point is required to compute (b_{eq}/x). The calculation is made by repeating the above sequence of operations (c) through (i) with x^* incremented by dx.

(j) Calculate the streamwise pressure gradient parameter β_S from the following:

$$\beta_s(x^*) = \beta_s(x^* - 2dx) + F_{dx}\left(d\,\beta_s - \beta_s\right)_{x^*-2dx}$$

where

$$F_x = 71/(70 + 1/c_x)$$

$$c_x = 1 - \frac{\left[G_L\left(\frac{b_{eq}}{x}\right)_L\right]_{x^*-dx}}{\left[G_L\left(\frac{b_{eq}}{x}\right)\right]_{x^*}}$$

and

$$\left(d\beta_s\right)_{x^*-2dx} = \left[2\left(\frac{H}{i_e}\right)\left(\frac{b_{eq}}{x}\right)_L\right]_{x^*-2dx} \qquad u_e = 0$$

$$= \left[2\left(\frac{H}{i_e}\right)\left(\frac{b_{eq}}{x}\right)_L \frac{x^*}{u_e}\frac{du_e}{dx}\right]_{x^*-2dx} \qquad u_e \neq 0$$

(k) Calculate the reference Reynolds number $R_{r,Q}$: calculate J_L [JAYELL] and $\mu_{o,}$ $\mathcal{L}$ [MUZERO]

then

$$\left(\frac{S_{eq}}{x}\right)_L = \frac{\left(b_{eq}/x\right)_L}{J_L^{\,2}} \quad ; \quad F_{x,Q} = \left[\frac{J_L^{\,17/20}\left(\frac{b_{eq}}{x}\right)_T}{\left(\frac{b_{eq}}{x}\right)_L}\right]^{1/3}$$

$$\left(\frac{S_{eq}}{x}\right)_T = \frac{\left(b_{eq}/x\right)_L}{J_L^{\,.9/16}} \quad ; \quad R_{r,Q} = \frac{\rho_r \mu_r u_e \left(x_{eq}/x\right)_L x^*}{\mu_o^{\,2} F_{x,Q}^{\,2}}$$

$$\left(\frac{x_{eq}}{x}\right)_L = \frac{\left(b_{eq}/x\right)_L}{J_L^{\,4/10}}$$

If $R_{r,Q} \geq R_{transition}$ (input) a new value of $\left(b_{eq}/x\right)_T$ is calculated using the following equations:

$$R_{L,S} = \frac{\rho_r \mu_r u_e S_{eq,L}}{\mu_o^{\,2}}$$

$$R_{T,S} = \frac{\rho_r \mu_r u_e S_{eq,T}}{\mu_o^{\,2}}$$

$$R_{\theta,L} = \frac{\mu_o}{\mu_e}\left[.664\left(R_{L,S}\right)^{1/2}\right] = \frac{\rho_e u_e \theta_L}{\mu_e}$$

$$R_{\theta,T} = \frac{\mu_o(.2135\ R_{T,S})}{\mu_e\left(\log_{10} R_{T,S} - .407\right)^{2.64}} = \frac{\rho_e u_e \theta_T}{\mu_e}$$

$$C_{transition} = \left[1 - \left(\frac{R_{\theta,L}}{R_{\theta,T}}\right)^{5/4}_{transition}\right]\int_0^{x_{transition}} G_T dx$$

$$\left(\frac{b_{eq}}{x}\right)_T = \left(\frac{b_{eq}}{x}\right)_T - \frac{C_{TR}}{G_T x^*}$$

If $R_{r,Q} < R_{transition}$ (input), a check is made at this point for x-printout values. If a printout point does not lie in the open interval ($x^* - 2dx$, $x^* + 2dx$), where x^* is the current value of x, the sequence (c) through (k) is repeated until one is encountered.

(l) When a printout point lies in the above interval and when the transition point is found the desired heat transfer data is calculated using the following equations:

Calculate J_L and $\mu_{o}, \mathcal{L}$ [JAYELL] and [MUZERO]

Then

$$\left(\frac{S_{eq}}{x}\right)_L = \frac{\left(b_{eq}/x\right)_L}{J_L^{\,2}}$$

$$\left(\frac{x_{eq}}{x}\right)_L = \frac{\left(b_{eq}/x\right)_L}{J_L^{\,4/10}}$$

$$F_{x,S} = \left[J_L^{17/4} \frac{\left(b_{eq}/x\right)_T}{\left(b_{eq}/x\right)_L} \right]^{1/3}$$

$$F_{x,Q} = \left[J_L^{17/20} \frac{\left(b_{eq}/x\right)_T}{\left(b_{eq}/x\right)_L} \right]^{1/3}$$

Reference Reynolds numbers:

$$R_{r,Q} = \frac{\rho_r \mu_r u_e \left(x_{eq}/x\right)_L x^*}{\mu_o^2 F_{x,Q}^2}$$

$$R_{r,S} = \frac{\rho_r \mu_r u_e \left(S_{eq}/x\right)_L x^*}{\mu_o^2 F_{x,S}^2}$$

$$\frac{\tau_T}{u_e} = \frac{\mu_o F_{x,S} J_L^{3/2}}{\left(S_{eq}/x\right)_L x^*} \left\{ \frac{.185\, R_{r,S}}{\left[\log_{10}(R_{r,S} + 3000)\right]^{2.584}} \right\}$$

$$\frac{\tau_L}{u_e} = .332 \frac{J_L^{3/2}\ \mu_o F_{x,S}\left(R_{r,S}\right)^{1/2}}{\left(S_{eq}/x\right)_L x^*}$$

Skin friction coefficients:

$$C_{f,e,L} = 2 \frac{\tau_L}{u_e} \frac{1}{\rho_e u_e}$$

$$C_{f,e,T} = 2 \frac{\tau_T}{u_e} \frac{1}{\rho_e u_e}$$

where

$$\rho_e = 1.232 \frac{P/P_{SL}}{ZT_e}$$

Heat transfer coefficients based on enthalpy:

$$H_L = \frac{.332\, g \mathcal{L} J_L^{3/10}}{\sigma_r^{.645}} \; \frac{\mu_o F_{x,Q} \left(R_{r,Q}\right)^{1/2}}{\left(x_{eq}/x\right)_L x^*}$$

$$H_T = \frac{g \mathcal{L} J_L^{3/10} \mu_o F_{x,Q}}{\sigma_r^{.645} \left(x_{eq}/x\right)_L x^*} \left\{ \frac{.185\, R_{r,Q}}{\left[\log_{10}\left(R_{r,Q} + 3000\right)\right]^{2.584}} \right\}$$

$$H_L = \frac{H_L}{H_o}$$

$$H_T = \frac{H_T}{H_{ref,T}}$$

Nonisothermal wall parameters:

$$\Phi_{local} = \left(i_{w_{local}} - i_{w_o}\right) + \sum_{i=1}^{n} \left(i_{w_i} - i_{w_{i-1}}\right) F_{\bar{S}_i}$$

where

$$F_{\bar{S}_i} = \left[1 - \left(\bar{S}_L^*\right)_i^{3/4}\right]^{-1/3} \qquad \text{laminar}$$

$$F_{\bar{S}_i} = \left[1 - \left(\bar{S}_T^*\right)_i^{9/10}\right]^{-1/9} \qquad \text{turbulent}$$

and

$$\bar{S}_{L_i}^* = \frac{S_{L_i} - \frac{1}{3}\left(S_{L_i} - S_{L_{i-1}}\right)}{\left(S_L\right)_{local}}; \quad S_{L_i} = \int_0^{x_i} G_L dx$$

$$\bar{S}_{T_i}^* = \frac{S_{T_i} - \frac{1}{3}\left(S_{T_i} - S_{T_{i-1}}\right)}{\left(S_T\right)_{local}}; \quad S_{T_i} = \int_0^{x_i} G_T dx$$

Adiabatic wall enthalpy:

$$i_{aw,L} = i_e + \sigma_r^{1/2}(H_o - i_e)$$

$$i_{aw,T} = i_e + \sigma_r^{1/2}(H_o - i_e)$$

Nonisothermal heat transfer rates:

$$\dot{q}_L = H_L\left(i_{aw,L} - i_w + \Phi_{L_{local}}\right)$$

$$\dot{q}_T = H_T\left(i_{aw,T} - i_w + \Phi_{T_{local}}\right)$$

Isothermal heat transfer rates:

$$\dot{q}_{L,iso} = H_L\left(i_{aw,L} - i_w\right)$$

$$\dot{q}_{T,iso} = H_T\left(i_{aw,T} - i_w\right)$$

(m) Print out heat transfer data.

In order for the sequence of calculations to reach this point, at least one printout point x_{PO} must lie in the interval $(x^* - 2dx), (x^* + 2dx)$ where $x^* = x_i + (2k)dx$ (INTEGER $k \geq 0$) is the current value of x. Answers of interest are now printed using linear interpolation for those printout points satisfying $x^* - 2dx < x_{PO} \leq x^*$. If a printout point satisfies the criterion $x^* < x_{PO} < (x^* + 2dx)$ then all quantities are saved for the current value of x to use for the interpolation scheme at x + dx. Steps (a) through (m) are repeated until all printout points have been exhausted.

Input

Streamline Functions

XSTAR	x^*	streamline coordinate at dx intervals, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
TW	$T_w(x^*)$	wall temperature, °R
AIEE	$i_e(x^*)$	edge enthalpy, ft^2/sec^2

THETAS	$\theta_e(x^*)$	streamline edge angle, radians
DODI	$\Delta/\Delta_i(x^*)$	divergence parameters
RORI	$r/r_i(x^*)$	divergence parameters
UE	$u_e(x^*)$	edge velocity, ft/sec
ZTEX	$ZT_e(x^*)$	edge compressibility-temperature product, °R
OMEGAE	$\omega_e(x^*)$	edge viscosity temperature ratio, slug/ft-sec-°R

Initial and Reference Conditions

H	H	total enthalpy, ft^2/sec^2
XI	x_I	initial streamline value, ft
RTRANS	$R_{transition}$	transition Reynolds number
SIGR	σ_r	partial Prandtl evaluated at ZT_r
DX	dx	streamline coordinate increment, ft
XPO	x_{PO}	printout array, ft
KF	K_{PO}	number of values in printout array
AYEWO	i_{w_o}	reference wall enthalpy, ft^2/sec^2 (needed <u>only</u> if $T_w \neq$ constant)
BEQXLI	$\left(\frac{b_{eq}}{x}\right)_{x_I}$	initial value of b_{eq}/x (laminar and turbulent)
BEQXTI	$\left(\frac{b_{eq}}{x}\right)_{x_I}$	initial value of b_{eq}/x (laminar and turbulent)

<u>Output</u>

XSTAR	x^*	streamline coordinate, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
UE	$u_e(x^*)$	edge velocity, ft/sec
Δ/Δ_i	$\Delta/\Delta_i(x^*)$	distance between streamlines

RORI	$r/r_i(x^*)$	divergence parameter due to body-shock layer geometry
TW	$T_w(x^*)$	wall temperature, °R
ZTEX	$ZT_e(x^*)$	edge compressibility-temperature product, °R
OMEGAE	$\omega_e(x^*)$	edge viscosity temperature ratio, slug/ft-sec-°R
AIEE	$i_e(x^*)$	edge enthalpy, ft^2/sec^2
AYEW	$i_w(x^*)$	wall enthalpy, ft^2/sec^2
BEQXL	$b_{eq,L}/x(x^*)$	ratio of laminar equivalent distance parameter to distance x along streamline
BEQXT	$b_{eq,T}/x(x^*)$	ratio of turbulent equivalent distance parameter to distance x along streamline
AYEAWL	$i_{aw,L}(x^*)$	laminar adiabatic wall enthalpy, ft^2/sec^2
AYEAWT	$i_{aw,T}(x^*)$	turbulent adiabatic wall enthalpy, ft^2/sec^2
HL	$H_L(x^*)/H_o$	laminar heat transfer coefficient ratio, $lb_m/ft^2/sec$
HT	$H_T(x^*)/H_{ref,T}$	turbulent heat transfer coefficient ratio, $lb_m/ft^2/sec$
QLISO	$\dot{q}_{iso,L}(x^*)$	laminar isothermal heat transfer rate, Btu/ft^2-sec
QTISO	$\dot{q}_{iso,T}(x^*)$	turbulent isothermal heat transfer rate, Btu/ft^2-sec
QDOTL	$\dot{q}_L(x^*)$	laminar heat transfer rate, Btu/ft^2-sec
QDOTT	$\dot{q}_T(x^*)$	turbulent heat transfer rate, Btu/ft^2-sec
CFEL	$C_{f,e,L}(x^*)$	laminar skin friction coefficient
CFET	$C_{f,e,T}(x^*)$	turbulent skin friction coefficient
RRQ	$R_{r,Q}$	heat transfer reference Reynolds number
RRS	$R_{r,S}$	skin friction reference Reynolds number

19) REF

REF calculates the stagnation heat transfer coefficient based on a reference hemisphere of radius R_o; and the reference heat transfer coefficient based on a 60° infinite swept cylinder given the reference cylinder radius. The following procedure is used to calculate the stagnation conditions:

(a) Initialize $i_e = H$

(b) Obtain the following from the indicated subroutines:

i_{w_o}	[IWALL]
ZT_w, ZT_e, ZT_s, ω_w, ω_e, ω_s	[SOMEGA]
ω_r, $\rho_r \mu_r$, ZT_r, i_r, σ_r	[RORMUR]
J_L	[JAYELL]
μ_o, $\mathcal{L}$	[MUZERO]

(c) Calculate the stagnation heat transfer coefficient and heat transfer rate:

$$H_o = \frac{.664\, g\, \mathcal{L}}{\sigma_r^{.645}} \left[\frac{\rho_r \mu_r u_\infty J_L u'_{MN}}{R_{HEMI,\ ref}} \right]^{1/2}$$

$$\dot{q}_o = H_o (H - i_{w_o})$$

The reference conditions are calculated from the following:

(d) Reinitialize $M_N = 1/2\ M_\infty$

(e) Obtain the following from the indicated subroutine:

P_{ST}/P_∞, i_{ST}, u'_{MN}	[STACON]
$i_{w,\ ref}$	[IWALL]
ZT_w, ZT_e, ZT_s, ω_w, ω_e, ω_s	[SOMEGA]

ω_r, $\rho_r \mu_r$, ZT_r, i_r, σ_r [RORMUR]

Γ_c, Γ_o [EBAR]

μ_o, $\mathcal{L}$ [MUZERO]

(f) Calculate the following:

$$x_{eq, L} = \frac{.866\ R_{CYL, ref}}{(1 + \Gamma_c)\ u'_{MN}}$$

$$F_{x, Q} = \left[1.6\left(\frac{1 + \Gamma_o}{1 + .7625\ \Gamma_o}\right)\right]^{1/3} \frac{1 + \Gamma_o}{1 + \Gamma_c}$$

$$R_{r, Q, ref} = \frac{.866\ \rho_r \mu_r u_\infty x_{eq, L}}{\mu_o^2\ F_{x, Q}^2}$$

$$H_{ref, L} = \frac{.332\, g\, \mathcal{L}}{\sigma_r^{.645}} \frac{\mu_o F_{x, Q}}{x_{eq, L}} \left(R_{r, Q, ref}\right)^{1/2}$$

$$H_{ref, T} = \frac{.185\, g\, \mathcal{L}}{\sigma_r^{.645}} \frac{\mu_o F_{x, Q}}{x_{eq, L}} \frac{R_{r, Q, ref}}{\left[\log_{10}(R_{r, Q, ref} + 3000)\right]^{2.584}}$$

$$i_{aw, ref, L} = H - \left(1 - \sigma_r^{1/2}\right)\left(H - i_e\right)$$

$$i_{aw, ref, T} = H - \left(1 - \sigma_r^{1/3}\right)\left(H - i_e\right)$$

$$\dot{q}_{ref, L} = H_{ref, L}\ (i_{aw, L} - i_{w, ref})$$

$$\dot{q}_{ref, T} = H_{ref, T}\ (i_{aw, T} - i_{w, ref})$$

$$\tau_{ref, L} = \frac{.866\, u_\infty\, \sigma_r^{.645}}{g\,\mathcal{L}} H_{ref, L}$$

$$\tau_{ref, T} = \frac{.866\, u_\infty\, \sigma_r^{.645}}{g\,\mathcal{L}} H_{ref, T}$$

$$C_{f, c, ref} = \frac{\tau_{ref}}{q_\infty}$$

Input

PINF	P_∞	free stream pressure, lb/ft^2
ACH	M_∞	free stream Mach number
VEL	u_∞	free stream velocity, ft/sec
VELP	u'_{MN}	modified Newtonian velocity gradient, 1/sec
H	H	total enthalpy, ft^2/sec^2
POPSL	P_o/P_{SL}	stagnation pressure, atm
TWREF	$T_{w,\ ref}$	reference wall temperature, °R
RHMREF	$R_{HEMI, ref}$	reference hemisphere radius, ft
RCLREF	$R_{CYL, ref}$	reference cylinder radius, ft

Output

AYEWO	i_{w_o}	stagnation wall enthalpy, ft^2/sec^2
HO	H_o	stagnation heat transfer coefficient, lb_m/ft^2-sec
QDOTO	$\dot{q}_o$	stagnation heat transfer rate, Btu/ft^2-sec
HREFL	$H_{ref, L}$	laminar reference heat transfer coefficient, lb_m/ft^2-sec
HREFT	$H_{ref, T}$	turbulent reference heat transfer coefficient, lb_m/ft^2-sec

AYAWRL	$i_{aw, ref, L}$	laminar reference adiabatic wall enthalpy, ft^2/sec^2
AYAWRT	$i_{aw, ref, T}$	turbulent reference adiabatic wall enthalpy, ft^2/sec^2
TAURL	$\tau_{ref, L}$	laminar reference shear stress, lb/ft^2
TAURT	$\tau_{ref, T}$	turbulent reference shear stress, lb/ft^2
QREFL	$\dot{q}_{ref, L}$	laminar reference heat transfer rate, lb_m/ft^2-sec
QREFT	$\dot{q}_{ref, T}$	turbulent reference heat transfer rate, lb_m/ft^2-sec
CFREFL	$C_{f, e, ref, L}$	laminar reference skin friction coefficient
CFREFT	$C_{f, e, ref, T}$	turbulent reference skin friction coefficient
AYEREF	$i_{w, ref}$	reference wall enthalpy, ft^2/sec^2

20) RORMUR

RORMUR calculates the following reference properties:

(a) Viscosity parameter

$$K^* = \frac{\omega_s}{\omega_w}\left[\frac{1.005}{.005 + \left(\frac{\omega_s}{\omega_e}\right)^7}\right]^{1/14}$$

$$K = (K^*)^{\left[\frac{13}{16} + \frac{3}{16} e^{-\left(K^* \frac{\omega_e}{\omega_w}\right)}\right]}$$

$$F_1 = K \frac{\omega_e}{\omega_w}$$

$$\omega_r = \omega_w F_1^{7/8} \left(\frac{1.2}{.2 + F_1^5}\right)^{1/10}$$

(b) Density-viscosity product

$$P_r = \left(\frac{P}{P_{SL}}\right) P_{SL}$$

$$\rho_r \mu_r = P_r \frac{\omega_r}{R}$$

(c) Compressibility-temperature product ZT_r

Change units of ω_r to slug/ft-sec-°K

$$\omega_{r,\,°K} = 1.8\ \omega_{r,\,°R}$$

For $3.585 \times 10^{-10} \le \omega_r \le 8.03 \times 10^{-10}$

$$ZT_{r,\,°K} = \frac{23213.13 \times 10^{-20}}{\omega^2_{r,\,°K}} \left[1 + \left(1 - .478656 \times 10^{18}\ \omega^2_{r,\,°K}\right)^{1/2}\right]^2$$

Otherwise $ZT_{r,\,°K}$ is obtained from interpolation

$$ZT_{r,\,°K} = f\left(\log_{10} P,\ \omega_{r_x}\right) \qquad \text{[TABL 19]}$$

$ZT_{r,\,°K}$ is converted to °R by $ZT_{r,\,°R} = 1.8\ ZT_{r,\,°K}$

(d) Enthalpy i_r

$$\frac{i_r}{RT_o} = f\left(\log_{10} P,\ ZT_{r,\,°K}\right) \qquad \text{[TABL 20]}$$

$$i_r = 847559.8 \left(\frac{i_r}{RT_o}\right)$$

(e) Prandtl number σ_r

$$\sigma_r = f(ZT_r) \qquad \text{[TABL 14]}$$

Input

POPSLX	P/P_{SL}	local pressure along streamline, atm
OMEGS	ω_s	stagnation viscosity-temperature ratio, slug-ft-sec-°R
OMEGE	ω_e	edge viscosity-temperature ratio, slug/ft-sec-°R
OMEGW	ω_w	wall viscosity-temperature ratio, slug/ft-sec-°R

Output

OMEGR	ω_r	reference viscosity-temperature ratio, slug/ft-sec-°R
ROMUR	$\rho_r \mu_r$	reference density-viscosity product, $slug^2/ft^4$-sec
ZTR	ZT_r	reference compressibility-temperature product, °R
AYER	i_r	reference enthalpy, ft^2/sec^2
SIGR	σ_r	reference Prandtl number

21) SIRCH (Y, X, YA, XA, N, IO)

SIRCH uses TAB to perform single interpolation on a table of $y = f(x)$.

Input

X	x	independent variable
YA		dependent array
XA		independent array
N		number of values in XA and YA arrays
IO		order of interpolation

Output

Y	y	dependent variable

22) SOMEGA (ENTH, POPSLX, ZT, OMEGA)

SOMEGA calculates the viscosity-temperature ratio ω and the compressibility-temperature product ZT as functions of enthalpy and pressure.

(a) Convert enthalpy to Btu/lb_m

$$i(Btu/lb_m) = 25301 \quad i(ft^2/sec^2)$$

(b) Calculate ZT

For $i \le 180$

$$ZT = i/.432$$

For $i > 180$

$$ZT = f(\log_{10} P/P_{SL}, i) \qquad [TABLE2]$$

Calculate ω

$ZT \le 111$ $\qquad \omega = 14.454 \times 10^{-10}$

$111 < ZT \le 2000$ $\qquad \omega = \left(\frac{1.8\ ZT}{200}\right)^{1/2} \left(\frac{14.454}{1.8\ ZT + 200}\right) 14.45 \times 10^{-10}$

$ZT > 2000$ $\qquad \omega = f(\log_{10} P/P_{SL}, ZT) \qquad [TABL18]$

(c) Convert ZT and ω to °R

$$ZT_{°R} = 1.8\ ZT_{°K}$$

$$\frac{\omega_{slug}}{ft\text{-}sec\text{-}°R} = \frac{\omega}{1.8}$$

Input

ENTH	i	enthalpy
POPSLX	P/P_{SL}	local pressure along streamline, atm

Output

ZT	ZT	compressibility-temperature product, °R
OMEGA	ω	viscosity-temperature ratio, slug/ft-sec-°R

23) SONENT (ENTH, SONIC)

SONENT calculates the speed of sound as a function of free stream enthalpy. The result is obtained by linear interpolation on tabular arrays constructed from the equation:

$$a_\infty = \left(\gamma R T_\infty\right)^{1/2}$$

where

$$T_\infty = f(i_\infty / RT_o)$$

$$\gamma_\infty = f(T_\infty)$$

Input

ENTH	i_∞	free stream enthalpy, ft^2/sec^2

Output

SONIC	a_∞	speed of sound, ft/sec

24) STACON (PSTPNF, AYEST, VELP, ACH, ACHN, H, PINF, VEL)

STACON calculates the stagnation-free stream pressure ratio, the stagnation enthalpy, and the modified Newtonian pressure gradient.

(a) Stagnation enthalpy

$$i_{ST} = H - \frac{u_\infty^2}{2}\left[1 - \left(\frac{M_N}{M_\infty}\right)^2\right]$$

(b) Pressure ratio P_o/P_∞

$$\left(P_{ST}\right)_o = \frac{P_\infty}{P_{SL}}\left(1.2\, M_N^2\right)^{3.5}\left(\frac{6}{7M_N^2 - 1}\right)^{2.5} \qquad M_N \geq 1$$

$$\left(P_{ST}\right)_o = \frac{P_\infty}{P_{SL}}\left(1 + \frac{M_N^2}{5}\right)^{3.5} \qquad M_N < 1$$

$$\frac{\rho_2}{\rho_1} = \frac{6 M_N^2}{M_N^2 + 5}$$

Interpolate for

$$\left(ZT_{ST}\right)_o = f\left(P_{ST_o}, i_{ST}\right) \qquad \text{[SOMEGA]}$$

$$\left(P_{ST}\right)_1 = \frac{P_\infty}{P_{SL}} + \left(P_{ST_o} - \frac{P_\infty}{P_{SL}}\right)\left[\frac{2\left(\frac{\rho_2}{\rho_1}\right) - \frac{6006\left(ZT_{ST}\right)_o}{i_{ST}}}{2(\rho_2/\rho_1) - 1}\right]$$

Interpolate for

$$\left(ZT_{ST}\right)_1 = f(P_{ST}, i_{ST}) \qquad \text{[SOMEGA]}$$

$$\left(P_{ST}\right)_2 = \frac{P_\infty}{P_{SL}} + \left(P_{ST_o} - \frac{P_\infty}{P_{SL}}\right)\left[\frac{2(\rho_2/\rho_1) - \frac{6006\, ZT_{ST_1}}{i_{ST}}}{2(\rho_2/\rho_1) - 1}\right]$$

$$\frac{P_o}{P_\infty} = \frac{\left(P_{ST}\right)_o}{(P_\infty/P_{SL})}$$

(c) Modified Newtonian pressure gradient

$$u'_{MN_o} = \left\{\frac{2}{7}\left(1 + \frac{5}{M_N^2}\right)\left(1 - \frac{P_\infty}{P_o}\right)\left[\frac{6006\left(ZT_{ST}\right)_1}{i_{ST}}\right]\right\}^{1/2}$$

Interpolate to find the empirical correction factor to the modified Newtonian velocity gradient as a function of $M_\infty \cos\Lambda$ (Swept cylinder) or M_∞ (Hemisphere).

$$u'_{MN} = (u'_{MN}) \cdot (\text{CORRECTION FACTOR})^5$$

For the correction factor see Figure 4.

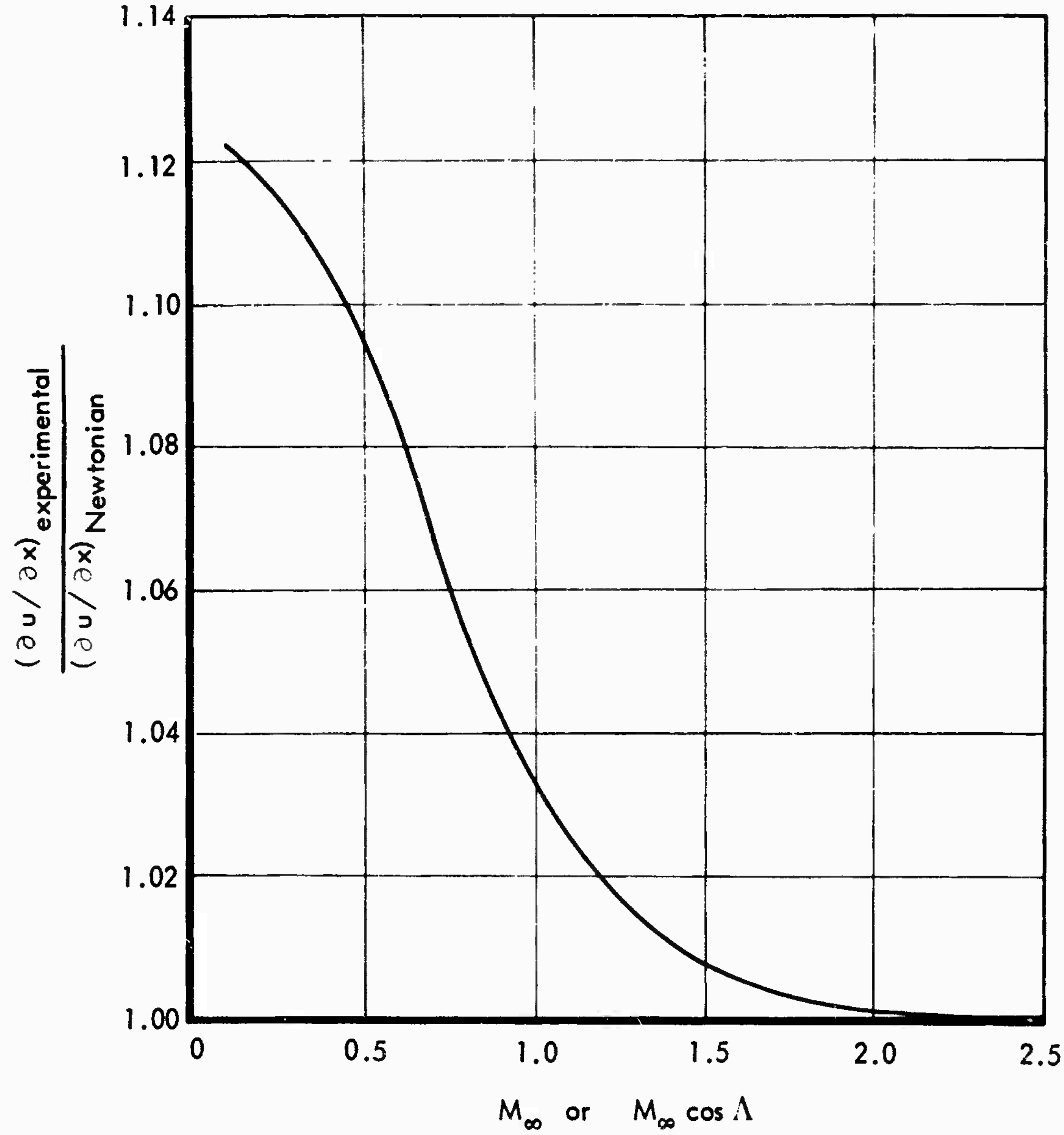


Figure 4: CORRECTION FACTOR FOR THE VELOCITY GRADIENT AT A HEMISPHERE STAGNATION POINT OR CYLINDER STAGNATION LINE

Input

ACH	M_∞	free stream Mach number
ACHN	M_N	normal Mach number
H	H	total enthalpy
PINF	P_∞	free stream pressure, lb/ft^2
VEL	u_∞	free stream velocity, ft/sec

Output

PSTPNF	P_0/P_∞	stagnation to free stream pressure ratio
AYEST	i_{ST}	stagnation enthalpy, ft^2/sec^2
VELP	u'_{MN}	modified Newtonian velocity gradient, 1/sec

25) STREAM (Program C)

STREAM calculates the streamline coordinate x as a function of the geometry coordinates ξ and η. A complete description of the geometry and streamline pattern are provided by the coordinates ξ and η, $\theta_e(\xi, \eta)$, $\eta_{max}(\xi)$ and $\eta_{max}(\xi)$.

$\theta_e(\xi, \eta)$ must be provided for the grid of ξ and η that lies within the geometry boundaries. With $\theta_{max}(\xi)$ known along the upper η geometry boundary the complete streamline pattern on the body is defined. $\eta_{max}(\xi)$ specifies the upper η boundary thus defining the geometry.

The coordinates (ξ_f, η_f) determine a point on a particular streamline. The streamline calculation begins at (ξ_f, η_f) and proceeds upstream until the geometry boundary is reached. The calculation of the streamline is done by the following procedure:

Let $(\xi_x)_1 = \xi_f$, $(\eta_x)_1 = \eta_f$, and $(x)_1 = 0$. Linear interpolation is performed on $\theta_e(\xi, \eta)$ and $\theta_{max}(\xi)$ if necessary, to determine $(\theta_x)_1$. Then

$$\left(d\eta_x\right)_1 = dx \sin\left(\theta_x\right)_1$$

$$\left(d\xi_x\right)_1 = dx \cos\left(\theta_x\right)_1$$

Then the coordinates of the new point are:

$$\left(\xi_x\right)_2 = \left(\xi_x\right)_1 - \left(d\xi_x\right)_1$$

$$\left(\eta_x\right)_2 = \left(\eta_x\right)_1 - \left(d\eta_x\right)_1$$

$$\left(x_2\right)_2 = dx$$

Continuing this procedure, the general equations are.

$$\left(\theta_x\right)_{K-1} = f\left[\left(\xi_x\right)_{K-1}, \left(\eta_x\right)_{K-1}\right]$$

$$\left(d\xi_x\right)_{K-1} = dx \sin\left(\theta_x\right)_{K-1}$$

$$\left(d\eta_x\right)_{K-1} = dx \cos\left(\theta_x\right)_{K-1}$$

and

$$\left(\xi_x\right)_K = \xi_F - \sum_{i=1}^{K-1} \left(d\xi_x\right)_i$$

$$\left(\eta_x\right)_K = \eta_F - \sum_{i=1}^{K-1} \left(d\eta_x\right)_i$$

$$(x)_K = \sum_{i=1}^{K-1} d\left(x_x\right)_i$$

The streamline calculation is completed by rearranging the ξ_x, η_x and x arrays such that ξ_x, η_x and x are measured from the upstream end of the streamline.

The subroutine STREAM assumes zero streamline divergence due to body geometry

$$\frac{r}{r_i} = 1.0$$

Input

M		number of ξ coordinates
N		number of η coordinates
XSIF	ξ_f	final streamline ξ coordinate, ft
ETAF	η_f	final streamline of η coordinate, ft
DX	dx	incremental streamline distance, ft
XSI	ξ	geometry coordinate, ft
ETA	η	geometry coordinate, ft
ETAMAX	η_{max}	body edge coordinate, ft
TH8MX	θ_{max}	streamline angle at body edge, degrees
THETAE	θ_e	streamline angle, degrees

Output

II		number of points calculated along streamline
XSIX	ξ_x	streamline coordinate, ft
ETAX	η_x	streamline coordinate, ft
X	x	streamline coordinate, ft
THETAX	θ_x	streamline angle, degrees
XF	x_f	final x-coordinate on streamline, ft

26) TABLE2 (AYES, PEDGEL, TS, CHECK)

TABLE2 obtains by table interpolation the compressibility-temperature product as a function of pressure and enthalpy.

Input

AYES	i	enthalpy, Btu/lb_m
PPLOG	$\log_{10} P/P_{SL}$	log of ratio of pressure to sea level pressure, pressure in atm

Output

TS	ZT	compressibility-temperature product, °R
CHECK		program check to determine success of interpolation (if the interpolation is not successful, an error statement is printed)

27) TABLE6 (TW, PPLOG, AYEW)

TABLE6 interpolates to find wall enthalpy as a function of pressure and wall temperature tables.

Input

TW	T_w	wall temperature, °K
PPLOG	$\log_{10} P/P_{SL}$	log of ratio of pressure to sea level pressure, pressure in atm

Output

AYEW	i_w	wall enthalpy, Btu/lb_m

28) TABLE7 (AYEE, PPLOG, AYEAYE)

TABLE7 interpolates on a table of pressure and edge enthalpy to find the dissociation on reaction enthalpy ratio i_D/i_e.

Input

AYEE	i_e	edge enthalpy, ft^2/sec^2
PPLOG	$\log_{10} P/P_{SL}$	log of ratio of pressure to sea level pressure, pressure in atm

Output

AYEAYE	i_D/i_e	dissociation reaction enthalpy ratio

29) TABL14 (ZTW, SIGT)

TABL14 obtains by table interpolation the partial Prandtl number as a function of the compressibility-temperature product.

Input

ZTW	ZT	compressibility-temperature product, °R

Output

SIGT	σ	Prandtl number

30) TABL18

TABL18 determines by table interpolation the viscosity-temperature ratio as a function of the pressure and compressibility-temperature product.

Input

ZT	ZT	compressibility-temperature product, °K
PPLOG	$\log_{10} P/P_{SL}$	log of ratio of pressure to sea level pressure, pressure in atm

Output

OMEGA	ω	viscosity-temperature ratio, slug/ft-sec-°K
CHECK		program check to determine success of interpolation (if interpolation is unsuccessful, an error comment is printed)

31) TABL19

TABL19 performs table interpolation to determine the compressibility-temperature product as a function of pressure and viscosity-temperature ratio. The table is composed of data in TABLE2 and TABL18.

Input

OMEGA	ω	viscosity-temperature ratio, slug/ft-sec-°K
PPLOG	$\log_{10} P/P_{SL}$	log of ratio of pressure to sea level pressure, pressure in atm

Output

ZT	ZT	compressibility-temperature product, °K

32) TABL20

TABL20 performs table interpolation to find i/RT_0 as a function of pressure and compressibility-temperature product.

Input

ZT	ZT	compressibility-temperature product, °K
PPLOG	$\log_{10} P/P_{SL}$	log of ratio of pressure to sea level pressure, pressure in atm

Output

AYERTO	i/RT_0	dimensionless enthalpy

33) TBLP (XT, YT, X, NTAB, N)

TBLP is a linear single interpolation routine. To save search time in finding the location of the independent x value in the x table, the search is begun at the previously located x value.

Input

XT	independent variable table
YT	dependent variable table
X	independent variable
NTAB	number of values in the x table
N	location in x table at which the search is begun. This value should be set = 1 upon the first entry to this routine. Thereafter, it stores the location of the previous search.

3. INPUT-OUTPUT DESCRIPTION

a. Data Input Preparation

Data are input to this program via punched cards. The purpose of this section is to define the required input and the form it takes on the cards. Input sheets for each option or section of the program are shown. Use of this type of form greatly reduces the amount of effort required to obtain results from the program. Data may be key punched directly from the forms.

1) AXISYMMETRIC OR TWO-DIMENSIONAL BODIES (Option D-1)

Card	Columns	Format	Description
1	1-6	A6	Type of case: FLIGHT, WT1, or WT2. This card indicates the type of free-stream conditions to be input.
1	7-10	----	Leave blank.
1	11-72	10A6	Title or some description of case.

Card #2 will consist of only one of the following types of input: Flight, Wind Tunnel 1 (WT1), or Wind Tunnel 2 (WT2). Wind Tunnel 1 case should be used for tunnels operating in the ideal gas region, while Wind Tunnel 2 case should be used for high energy tunnels such as shock tubes.

FLIGHT

2	1-10	F10	Flight altitude, ft.
2	11-20	F10	Flight velocity, ft/sec.

WT1

2	1-10	F10	Free-stream Mach number.
2	11-20	F10	Free-stream static temperature, R.
2	21-30	F10	Free-stream static pressure, lb/ft^2.

WT2

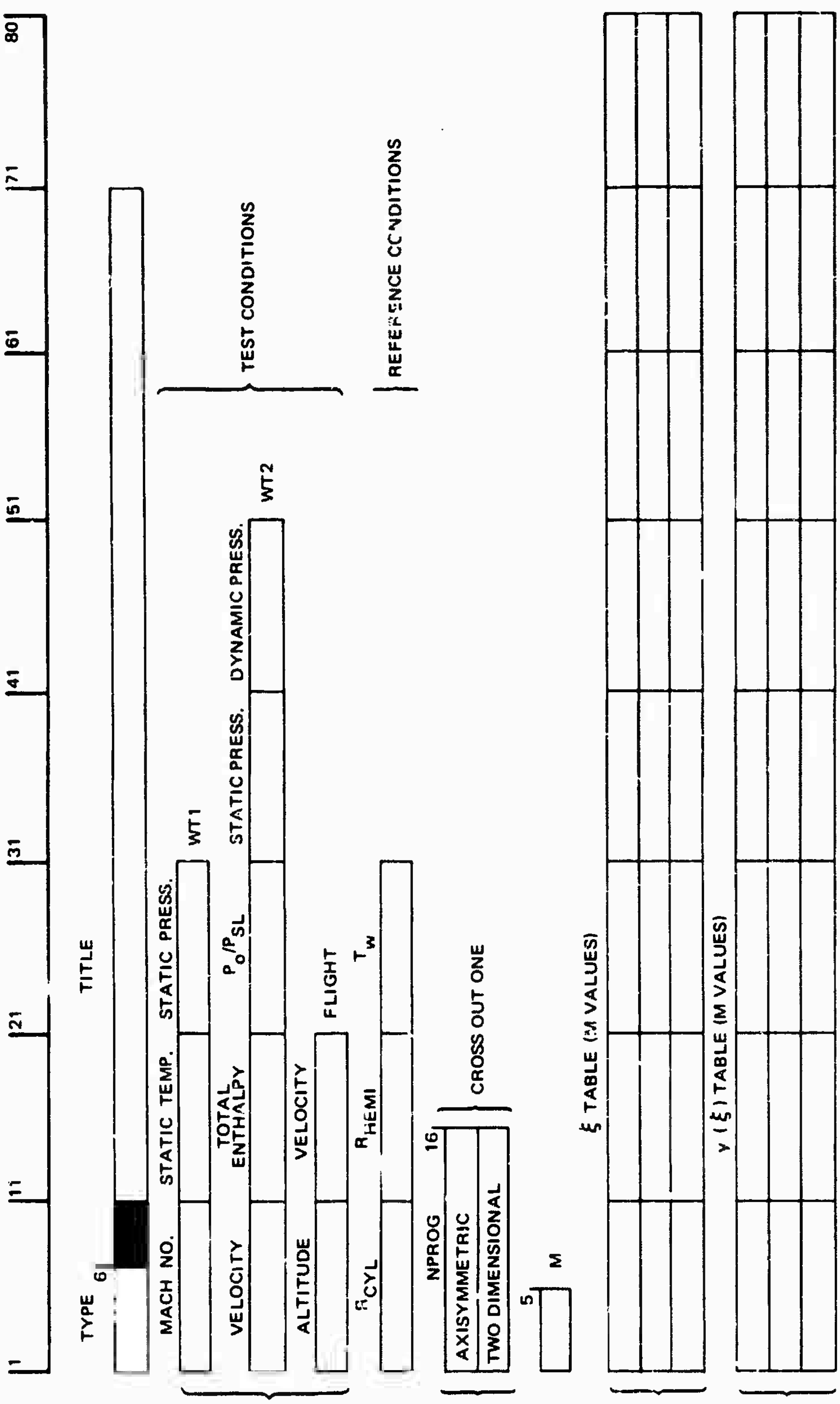
2	1-10	F10	Free-stream velocity, ft/sec.
2	11-20	F10	Total enthalpy, ft^2/sec^2.

Card	Columns	Format	Description
2	21-30	F10	Ratio of stagnation pressure to sea level pressure.
2	41-50	F10	Free-stream dynamic pressure, lb/ft^2.
3	1-10	F10	Radius of 60° swept infinite cylinder for the turbulent reference heating rate, ft.
3	11-20	F10	Radius of hemisphere for the laminar reference heating rate, ft.
3	21-30	F10	Reference wall temperature, °R.
4	1-14	A6	Contains the word AXISYMMETRIC or TWODIMENSIONAL.
5	1-5	I5	M is the number of values in the geometry tables $y(\xi)$ and ξ. $3 \leq M \leq 50$.
6	----	8F10	This table contains M values of ξ, ft.
7	----	8F10	This table contains M values of $y(\xi)$ corresponding to ξ in the preceding table, ft.

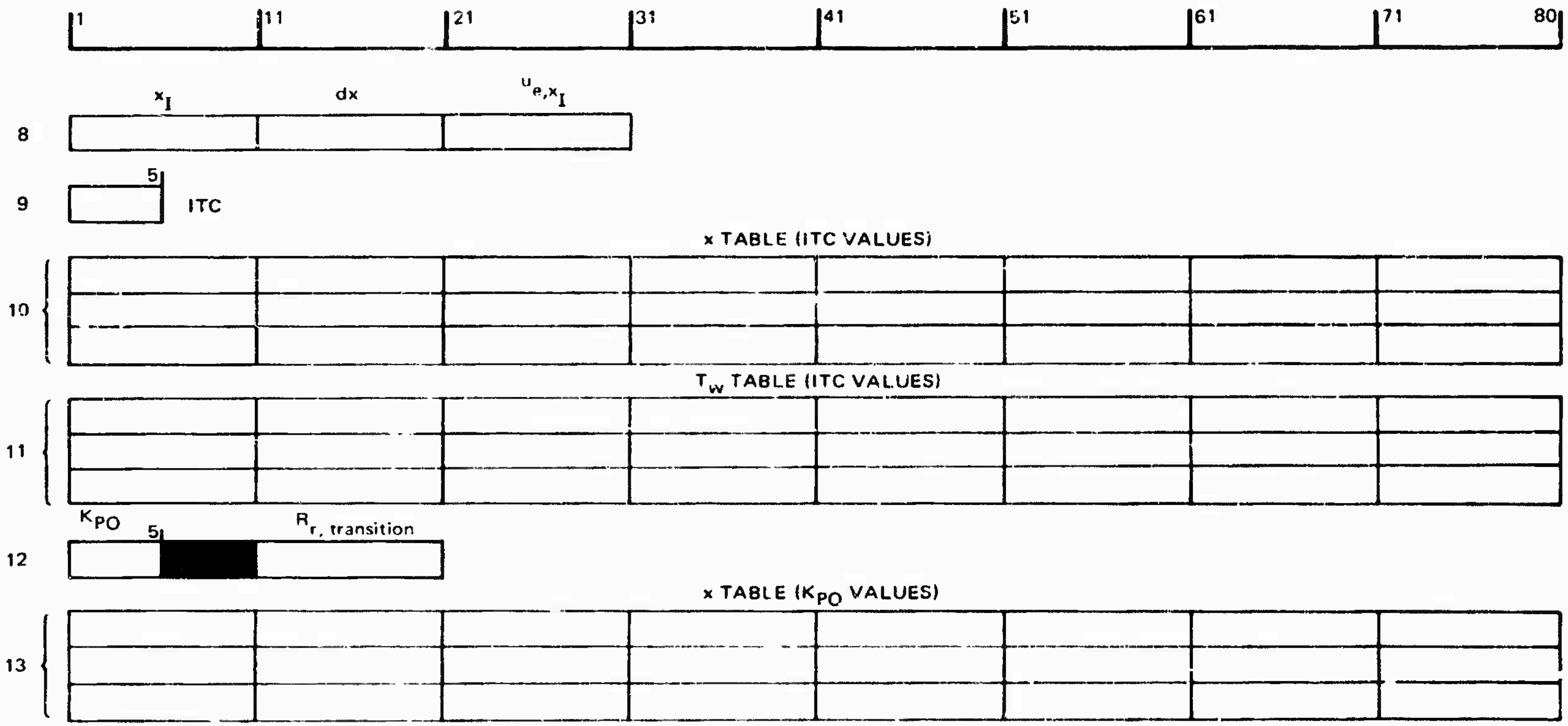
Note: A minimum of three points must be input into the preceding two tables. If the body is very slender, more points should be used.

Card	Columns	Format	Description
8	1-10	F10	Initial x location, point at which calculation begins, ft.
8	11-20	F10	Increment in x, ft. $dx > (x_f - x_o)/750$.
8	21-30	F10	Edge velocity at x_I, ft/sec.
9	1-5	I5	ITC is the number of values in the wall temperature versus surface distance tables. $2 \leq ITC \leq 50$. If $ITC \leq 3$ then the wall temperature must be nonisothermal.

Card	Columns	Format	Description
10	----	8F10	This table contains ITC values of surface distance, ft.
11	----	8F10	This table contains ITC values of wall or surface temperature corresponding to the preceding distance table, °R. If T_w = constant then enter two values.
12	1-5	I5	K_{PO} is the number of printout locations desired. $K_{PO} \leq 500$.
12	11-20	F10	Transition Reynolds number, see definition of $R_{r,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
13	----	8F10	This table contains K_{PO} locations at which printout is desired.

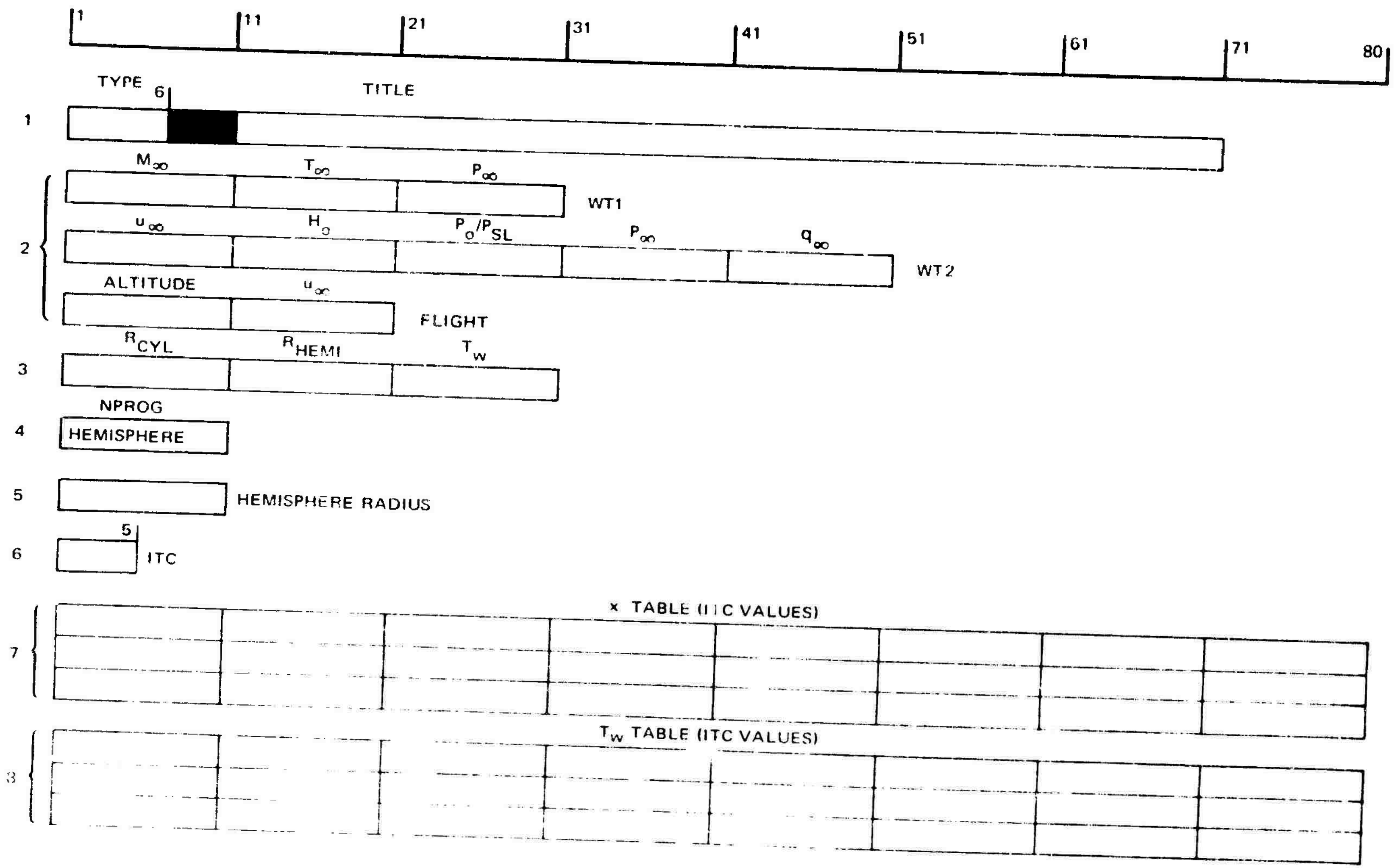


1
11
21
31
41
51
61
71
80
TYPE
6
TITLE
MACH NO.
STATIC TEMP.
STATIC PRESS.
WT1
VELOCITY
TOTAL ENTHALPY
P_o/P_{SL}
STATIC PRESS.
DYNAMIC PRESS.
WT2
TEST CONDITIONS
ALTITUDE
VELOCITY
FLIGHT
R_{CYL}
R_{HEMI}
T_w
REFERENCE CONDITIONS
NPROG
16
AXISYMMETRIC
TWO DIMENSIONAL
CROSS OUT ONE
5
M
ξ TABLE (M VALUES)
$y(\xi)$ TABLE (M VALUES)
1
2
3
4
5
6
7

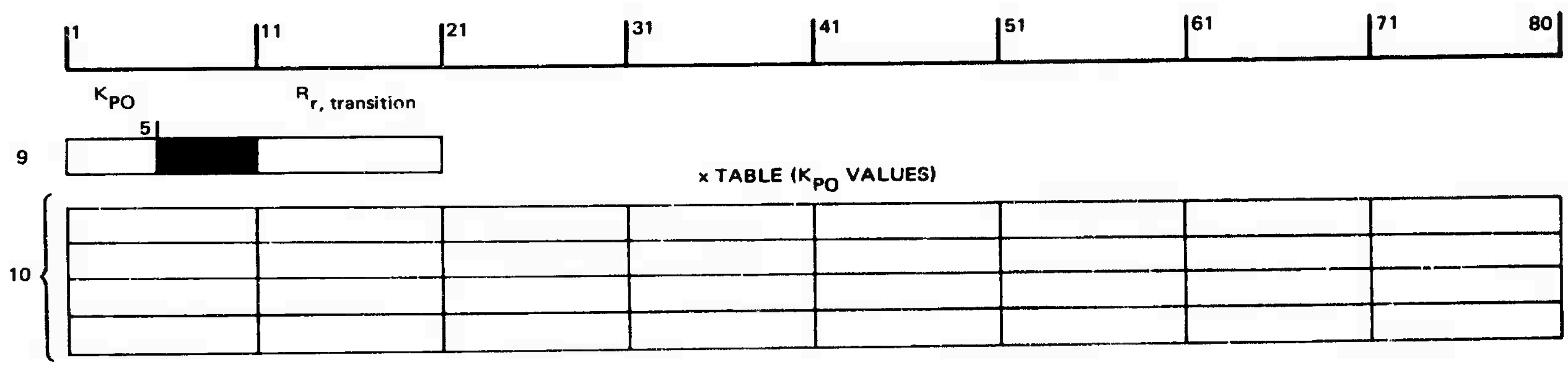

1
11
21
31
41
51
61
71
80
8
x_I
dx
u_{e,x_I}
9
5
ITC
10
x TABLE (ITC VALUES)
11
T_w TABLE (ITC VALUES)
12
K_{PO}
5
$R_{r,\ transition}$
13
x TABLE (K_{PO} VALUES)

2) HEMISPHERE CALCULATION (Option D-2)

Card	Columns	Format	Description
1	Same as Option D-1		
2			
3			
4	1-10	A6	Contains title of program - HEMISPHERE.
5	1-10	F10	Hemisphere radius, ft.
6	1-5	I5	ITC is the number of points in the T_W versus x tables. $2 \leq ITC \leq 50$. If $ITC \geq 3$ then the wall temperature must be nonisothermal.
7	----	8F10	This is a table of ITC values of x where T_W are input, ft.
8	----	8F10	This is a table of ITC values of T_W which correspond to locations given in the preceding x table, or if T_W = constant, enter two values of T_W.
9	1-5	I5	K_{PO} is the number of printout locations desired. $K_{PO} < 500$.
9	11-20	F10	Transition Reynolds number, see definition of $R_{r,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
10	----	8F10	This table contains K_{PO} surface distance locations around the hemisphere at which printout is desired.



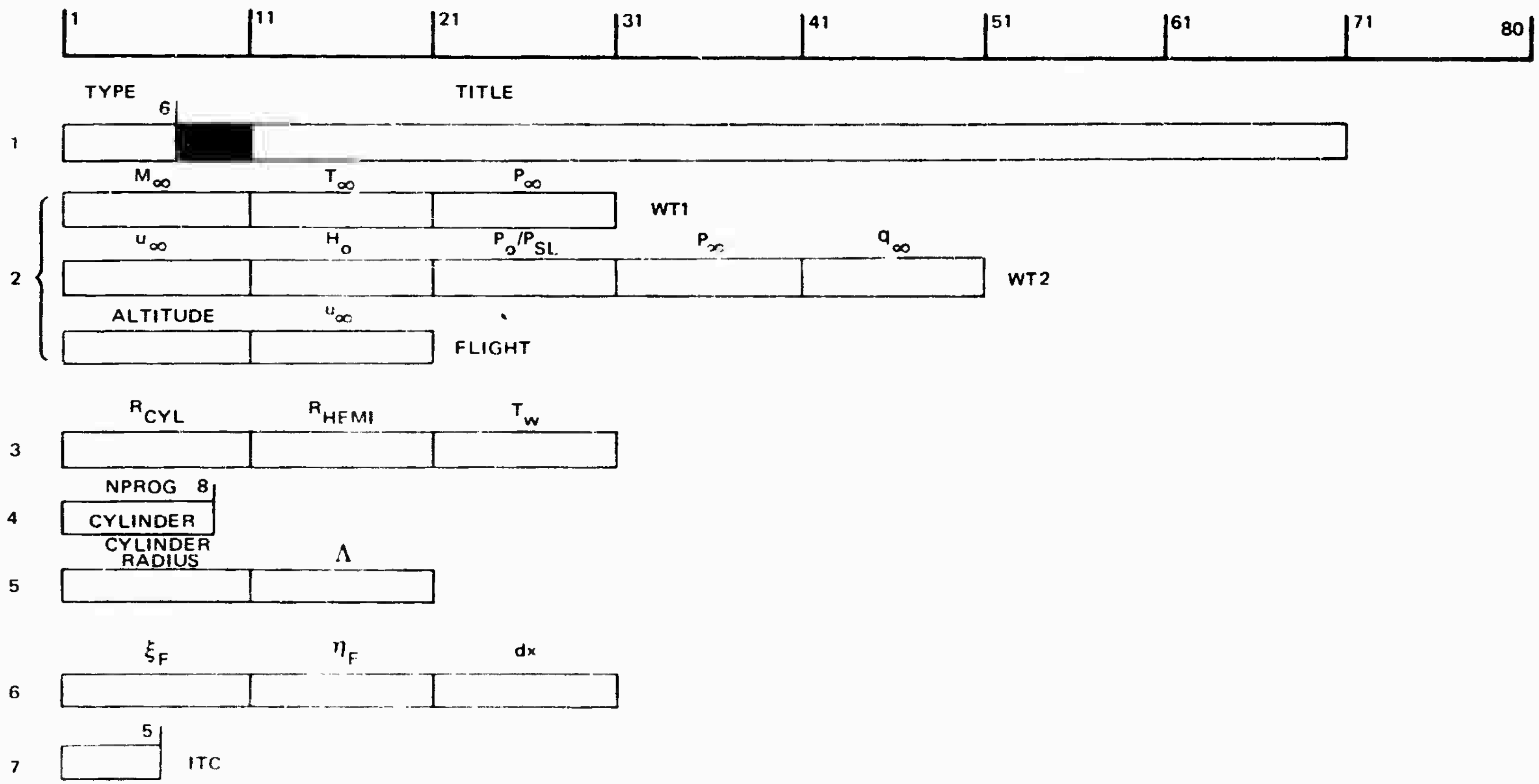
1
11
21
31
41
51
61
71
80
TYPE
6
TITLE
M_∞
T_∞
P_∞
WT1
u_∞
H_o
P_o/P_{SL}
P_∞
q_∞
WT2
ALTITUDE
u_∞
FLIGHT
R_{CYL}
R_{HEMI}
T_w
NPROG
HEMISPHERE
HEMISPHERE RADIUS
5
ITC
x TABLE (ITC VALUES)
T_w TABLE (ITC VALUES)
1
2
3
4
5
6
7
8

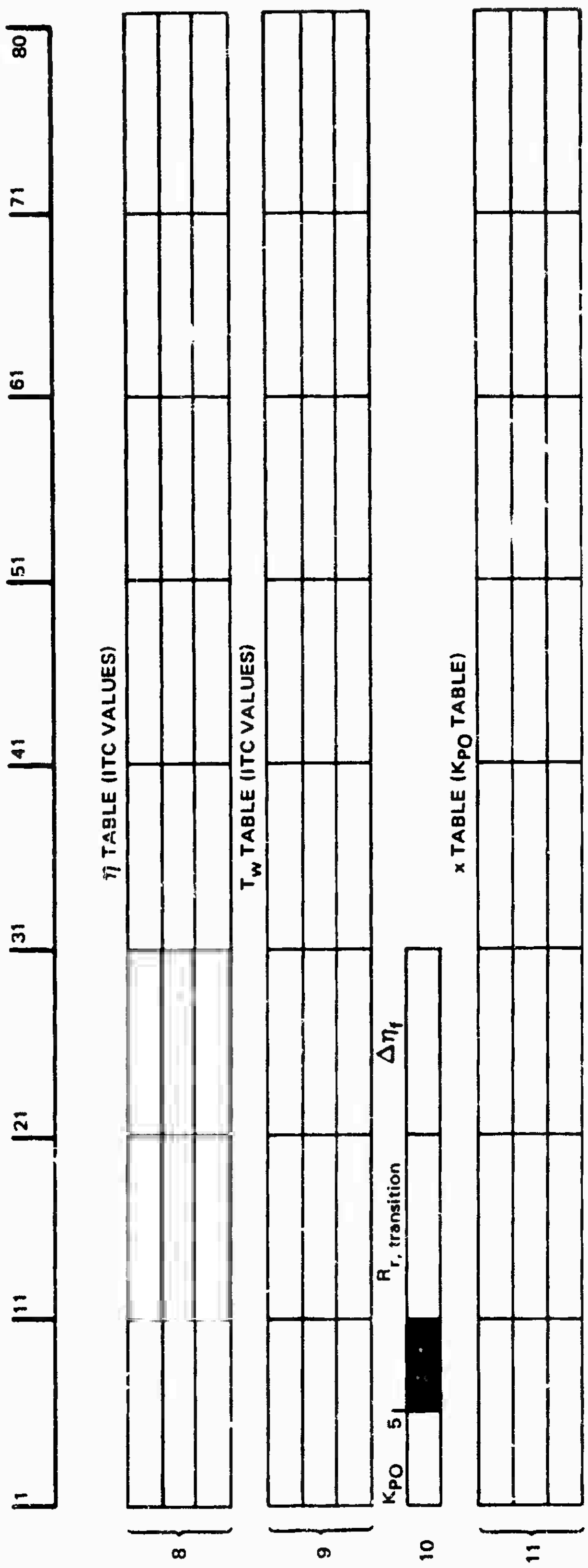

1
11
21
31
41
51
61
71
80
K_{PO}
$R_{r,\ transition}$
5
9
x TABLE (K_{PO} VALUES)
10

3) SWEPT INFINITE CYLINDER (Option D-3)

Card	Columns	Format	Description
1	Same as Option D-1		
2			
3			
4	1-8	A6	This card contains the word CYLINDER to allow the program to select the D-3 options.
5	1-10	F10	Cylinder radius. ft.
5	11-20	F10	Cylinder sweep angle. measured from a line normal to the flow. $0 < \Lambda < 90$.
6	1-10	F10	ξ_f is the abscissa at which the streamline calculation begins. ft. ξ_f must be large enough so that the streamline has very nearly zero slope at the stagnation line. Cases have been run by the authors using $\xi_f = 10$ and 20 ft for Λ = 40° and 60°.
6	11-20	F10	η_f is the ordinate at which the streamline calculation begins. ft.
6	21-30	F10	dx is the increment in distance along the streamline used in the boundary layer calculations (Program A). ft. $dx \geq (\xi_f^2 + \eta_f^2)^{1/2}/600$
7	1-5	I5	ITC is the number of values in T_w and η tables. $2 \leq ITC \leq 50$. If $ITC \geq 3$ the wall temperature is nonisothermal.
8	----	8F10	This table contains ITC values of η. where η is the distance around the cylinder, ft.
9	----	8F10	This table contains ITC values of T_w corresponding to the η table, °R.

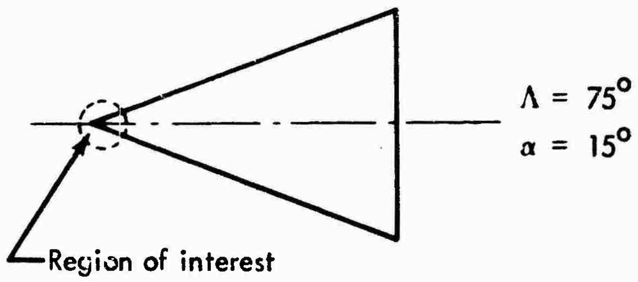
Card	Columns	Format	Description
10	1-5	I5	K_{PO} is the number of x printouts desired along the streamline. $K_{PO} < 500$.
10	11-20	F10	Transition Reynolds number, see definition of $R_{r,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
10	21-30	F10	$\Delta\eta_f$ is an increment used in Program C to locate the starting location of second streamline, ft. If $\Delta\eta_f$ is small, in some cases the resolution between streamlines near the streamlines origin or apex may be very poor. The poor resolution leads to heat transfer distributions that are not smooth. In the event this occurs $\Delta\eta_f$ should be increased, ft.
11	----	8F10	x_{PO} is a table containing locations along the streamline at which printout is desired.


1
11
21
31
41
51
61
71
80
TYPE
6
TITLE
1
M_∞
T_∞
P_∞
WT1
u_∞
H_o
P_o/P_{SL}
P_∞
q_∞
WT2
2
ALTITUDE
u_∞
FLIGHT
R_{CYL}
R_{HEMI}
T_w
3
NPROG
8
CYLINDER
4
CYLINDER RADIUS
Λ
5
ξ_F
η_F
dx
6
5
ITC
7


1
11
21
31
41
51
61
71
80
η TABLE (ITC VALUES)
8
T_W TABLE (ITC VALUES)
9
K_{PO}
5
R_r, transition
$\Delta\eta_f$
10
x TABLE (K_{PO} TABLE)
11

4) SHARP DELTA WING (Option D-4)

Cards	Columns	Format	Description
1, 2, 3	Same as Option D-1		
4	1-6	A-6	Contains the word DELTA.
5	1-5	I5	ITC is an indicator for wall temperature gradient. If the wall is at constant temperature input a 1 and one value of wall temperature in table.
6	1-10	F10	Angle of attack, degrees.
6	11-20	F10	Wing sweep angle measured from a line normal to the flow, degrees.
6	21-30	F10	ξ_f is the final or end coordinate along the streamline at which heating data is desired, ft.
6	31-40	F10	η_f is the final or end coordinate along the streamline at which heating data is desired, ft.
6	41-50	F10	x_I is the point along the streamline at which the calculations begin and may have to be greater than zero as indicated in Figure 5. The resulting change in streamline curvature, due to interpolation, near the apex produces a discontinuity in the heating calculations, ft.
7	1-10	F10	$\Delta\xi$ is the increment in ξ which the program uses to setup an (ξ, η) array of streamline angles, velocity, and pressures, ft. $$\Delta\xi \geq \frac{\xi_f}{50}$$



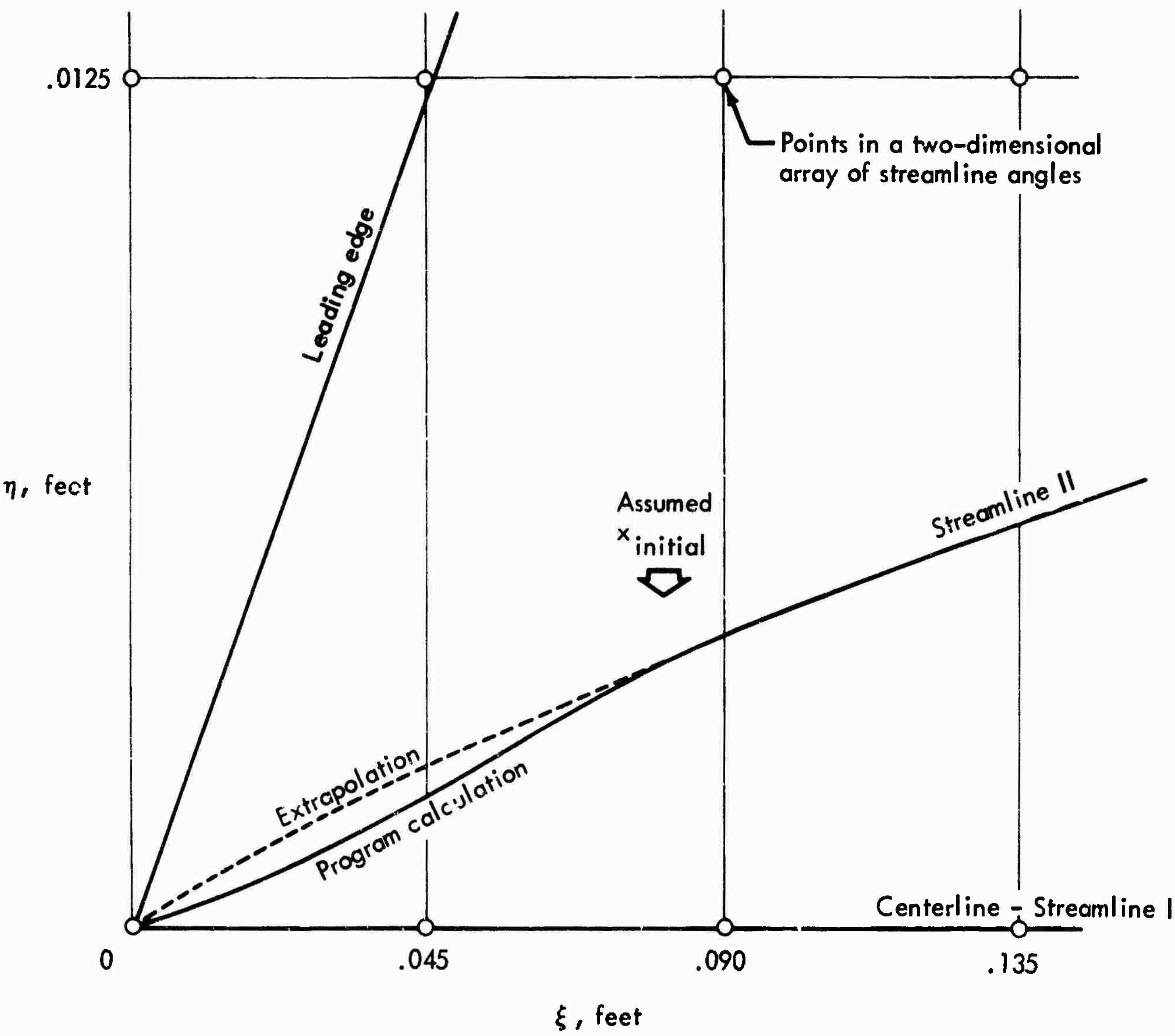
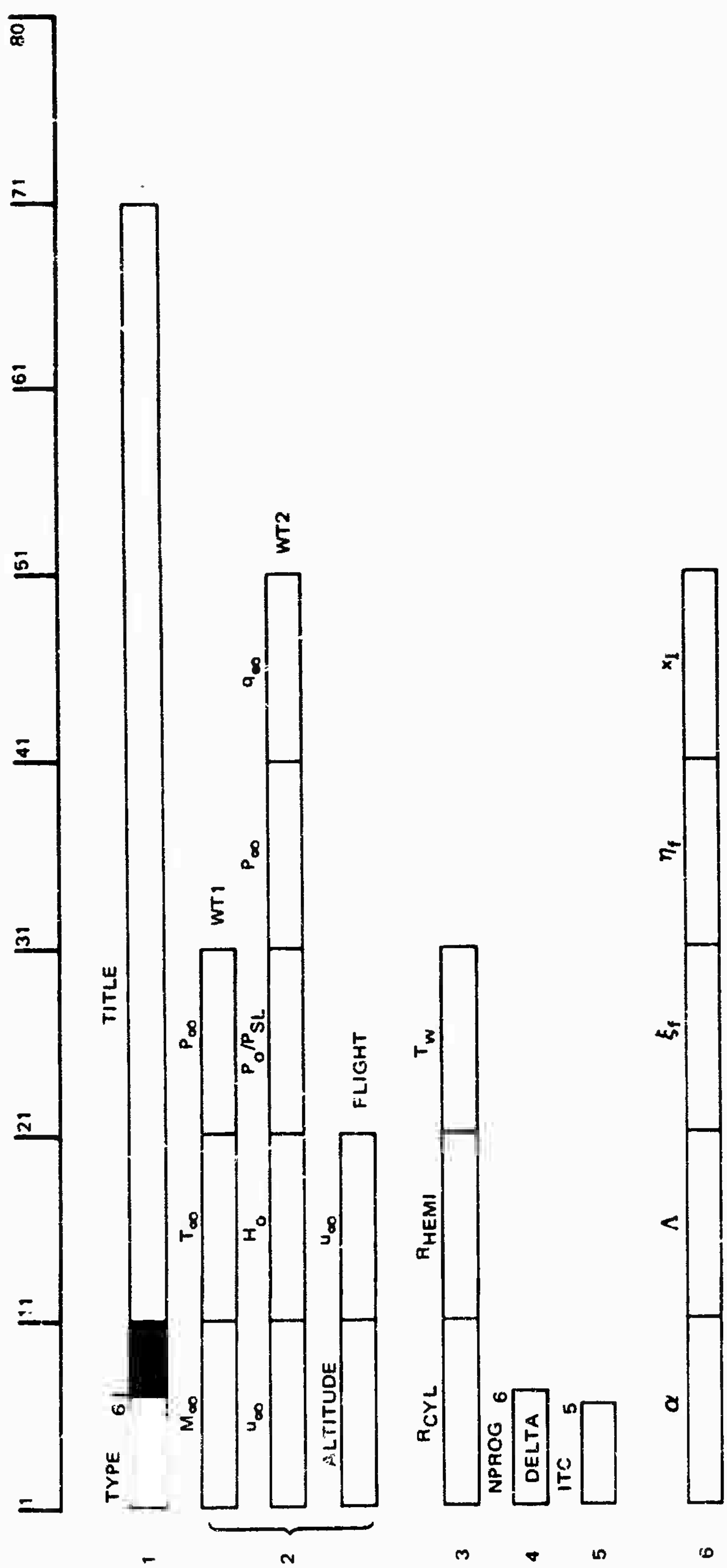


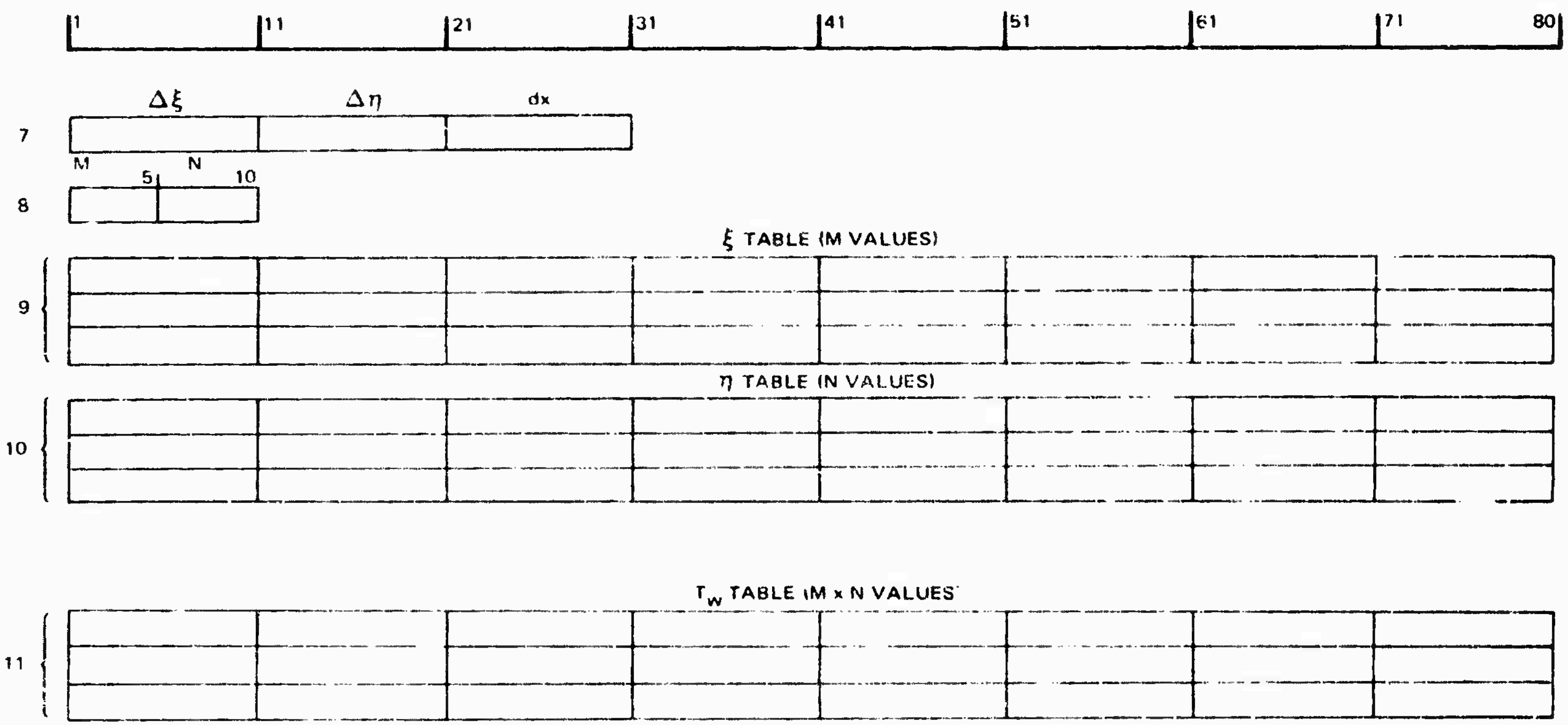
Figure 5: STREAMLINES NEAR THE APEX OF A SHARP DELTA WING

Card	Columns	Format	Description
7	11-20	F10	$\Delta\eta$ is the increment in η the program uses to setup an (ξ, η) array of streamline angles, velocity and pressures, ft. $$\Delta\eta \geq \frac{\eta_f}{50}$$
7	21-30	F10	dx is the increment in distance along the streamline at which calculations will be performed, ft. $dx > (x_f - x_I)/750$.

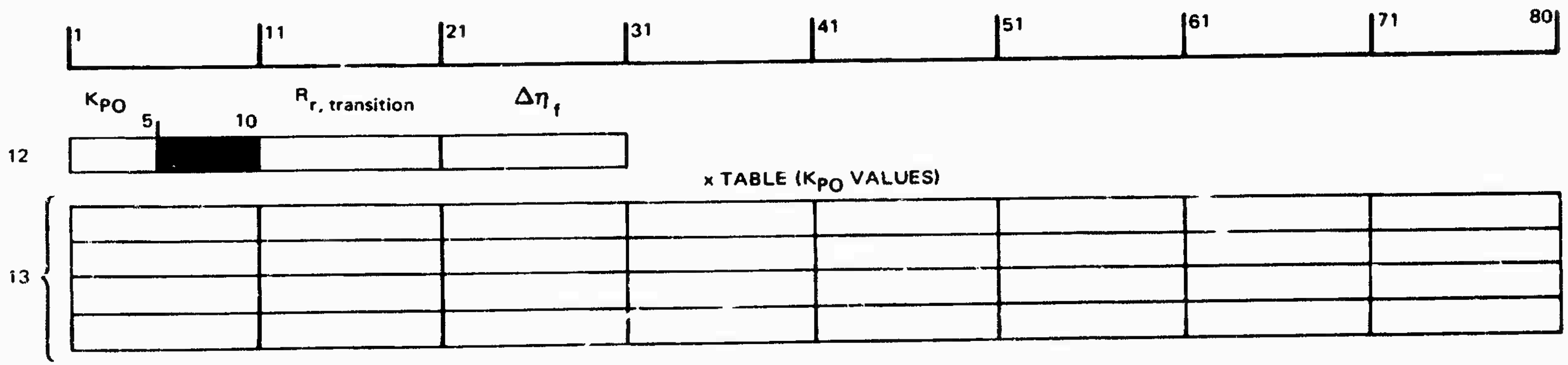
Note: If T_w is a constant value, enter one value into card 11, and delete cards 8, 9, and 10.

Card	Columns	Format	Description
8	1-5	I5	M is the number of points in the ξ array. $2 \leq M \leq 50$.
8	6-10	I5	N is the number of points in the η array. $2 \leq N \leq 50$.
9	----	8F10	This is a table of M values of ξ where wall temperature will be input, if ITC > 1.
10	----	8F10	This is a table of N values of η where wall temperature will be input if ITC > 1.
11	----	8F10	This is a table containing (N) $\cdot$ (M) values of wall temperature, input so ξ varies with each η.
12	1-5	I5	K_{PO} is the number of x printouts desired along the streamline. $K_{PO} < 500$.
12	11-20	F10	Transition Reynolds number, see definition of $R_{T,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
12	21-30	F10	See card 10, swept infinite cylinder, option D-3.
13	----	8F10	X_{PO} is a table of K_{PO} locations along the streamline where printout is desired.


1
11
21
31
41
51
61
71
80
TYPE 6
TITLE
M_∞
T_∞
P_∞
WT1
u_∞
H_o
P_o/P_{SL}
P_∞
q_∞
WT2
ALTITUDE
u_∞
FLIGHT
R_{CYL}
R_{HEMI}
T_w
NPROG 6
DELTA
ITC 5
α
Λ
ξ_f
η_f
x_I
1
2
3
4
5
6



1
11
21
31
41
51
61
71
80
$\Delta\xi$
$\Delta\eta$
dx
7
M
5
N
10
8
ξ TABLE (M VALUES)
9
η TABLE (N VALUES)
10
T_W TABLE (M x N VALUES)
11



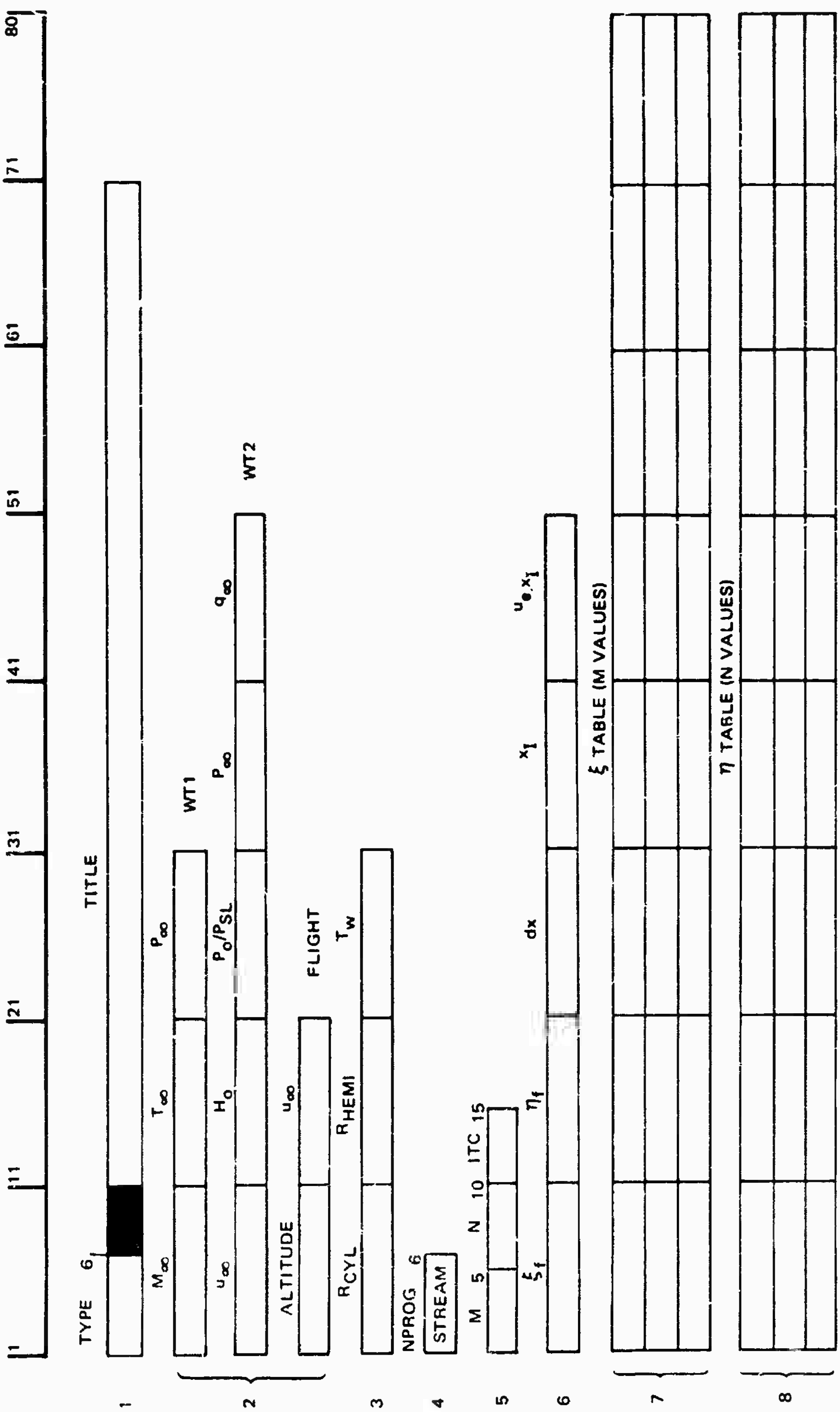
1
11
21
31
41
51
61
71
80
K_{PO}
5
10
$R_{r,\ transition}$
$\Delta\eta_f$
12
x TABLE (K_{PO} VALUES)
13

5) STREAMLINE CALCULATIONS (PROGRAM C)

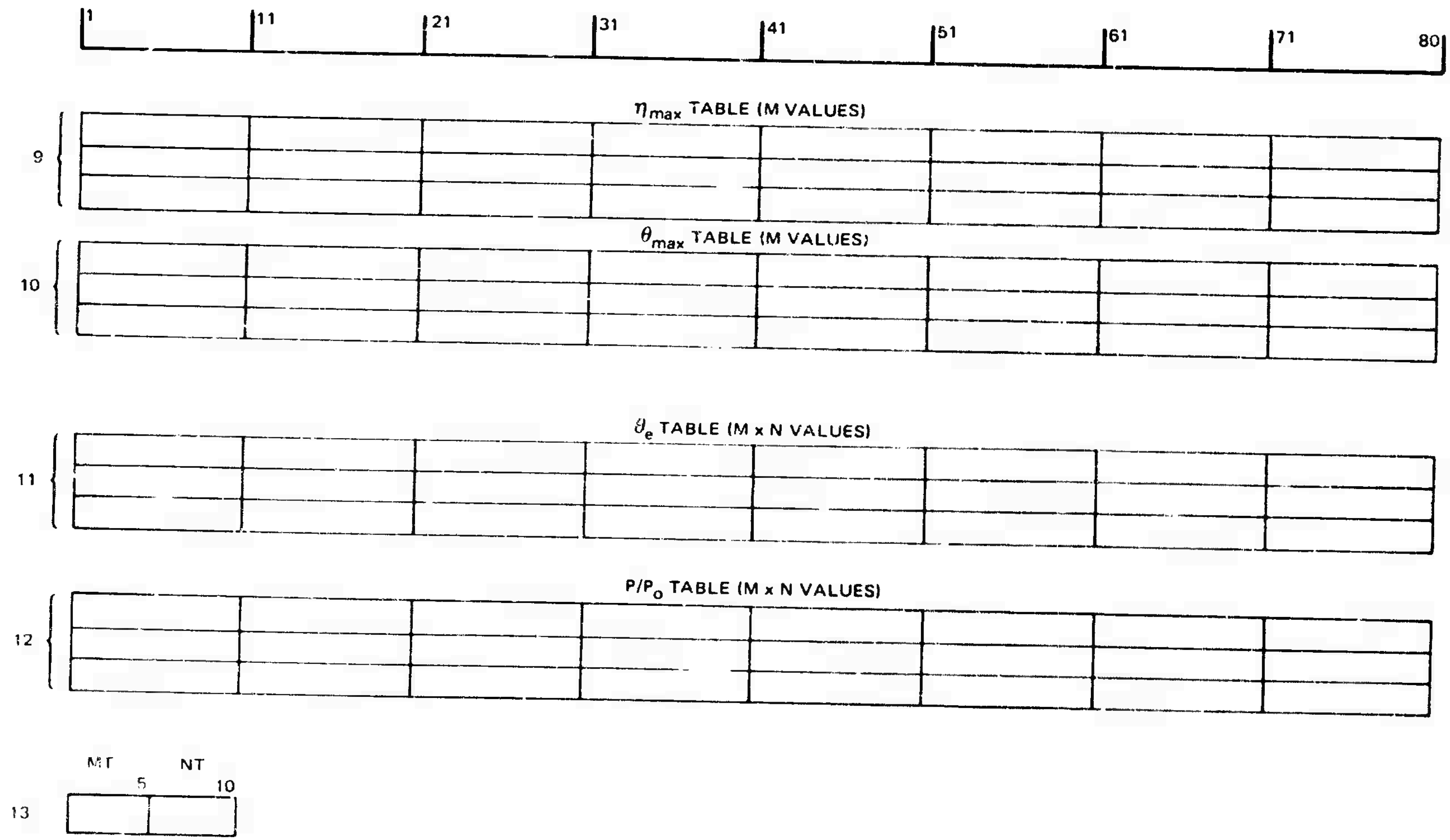
Card	Columns	Format	Description
1	Same as Option D-1		
2			
3			
4	1-6	A6	Must contain the word STREAM.
5	1-5	I5	M is the number of points in the array ξ. $2 \leq M \leq 50$.
5	6-10	I5	N is the number of points in the array η $2 \leq N \leq 50$.
5	11-15	I5	ITC is an indicator used to test whether the wall temperature is a constant or a variable. If T_w is a constant ITC = 1 and enter one value of T_w into card 16 and delete cards 13, 14, and 15.
6	1-10	F10	ξ_f is ξ at the final or end point, ft.
6	11-20	F10	η_f is η at the final or end point, ft.
6	21-30	F10	dx is the increment in distance along the streamline, ft. $dx \geq \frac{\left(\xi_f^2 + \eta_f^2\right)^{1/2}}{750}$
6	31-40	F10	x_I is the point along the streamline at which the calculations will begin, ft.
6	41-50	F10	u_{e,x_I} is the edge velocity at the initial or start of calculations, ft/sec.
7	----	8F10	This is a table of M values of ξ $2 \leq M \leq 50$. θ_e, P/P_0 and T_w will be input as functions of ξ and η, ft.

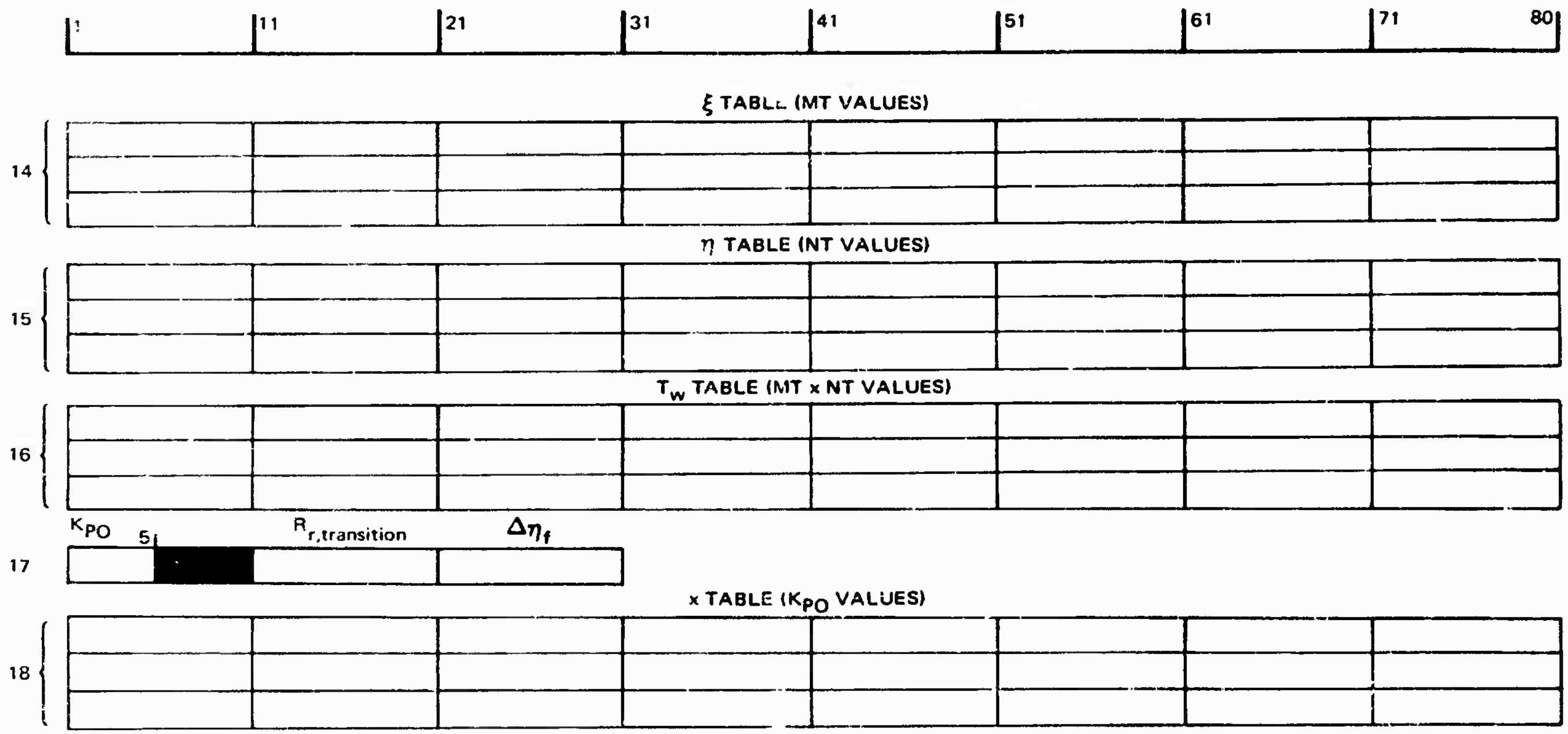
Card	Columns	Format	Description
8	----	8F10	This is a table of N values of η $2 \leq M \leq 50$. θ_e, P/P_o and T_w will be input as functions of ξ and η, ft.
9	----	8F10	This is a table of M values of η_{max} as a function of ξ. η_{max} defines the upper or outer boundary of the body, ft.
10	----	8F10	This is a table of M values of the streamline angle θ_{max} along the boundary as a function of ξ, deg.
11	----	8F10	This is a table of (M) · (N) values of the local streamline angle θ_e and should be input along lines of constant η and ξ, deg.
12	----	8F10	This is a table of (M) · (N) values the local pressure divided by the stagnation pressure and should be input along lines of constant η and varying ξ.
13	1-5	I5	MT is the number of ξ values entered into the table $T_w = f(\xi, \eta)$. Also see ITC above.
13	6-10	I5	NT is the number of η values entered into the table $T_w = f(\xi, \eta)$. Also see ITC above.
14	----	8F10	This is a table of MT values of ξ at which T_w will be input. Also see ITC above.
15	----	8F10	This is a table of NT values of η at which T_w will be input. Also see ITC above.
16	----	8F10	This is a table of (MT) · (NT) values of T_W as a function ξ and η, °R. T_W should be input as a function of ξ at constant η.
17	1-5	I5	K_{PO} is the number of x locations along the streamline at which printout is desired. $K_{PO} < 500$.

Card	Columns	Format	Description
17	11-20	F10	Transition Reynolds number. see definition of $R_{T.Q}$ in Program A. labeled QTRANS in section 2. b. 18.
17	21-30	F10	See card 10. Swept infinite cylinder. option D-3.
18	1-10	F10	xPO is a table of KPO locations along the streamline at which printout is desired.



1
11
21
31
41
51
61
71
80
TYPE 6
TITLE
M∞
T∞
P∞
WT1
u∞
H0
P0/PSL
P∞
q∞
WT2
ALTITUDE
u∞
FLIGHT
RCYL
RHEMI
TW
NPROG 6
STREAM
M 5
N 10
ITC 15
ξf
ηf
dx
xI
ue,xI
ξ TABLE (M VALUES)
η TABLE (N VALUES)
1
2
3
4
5
6
7
8

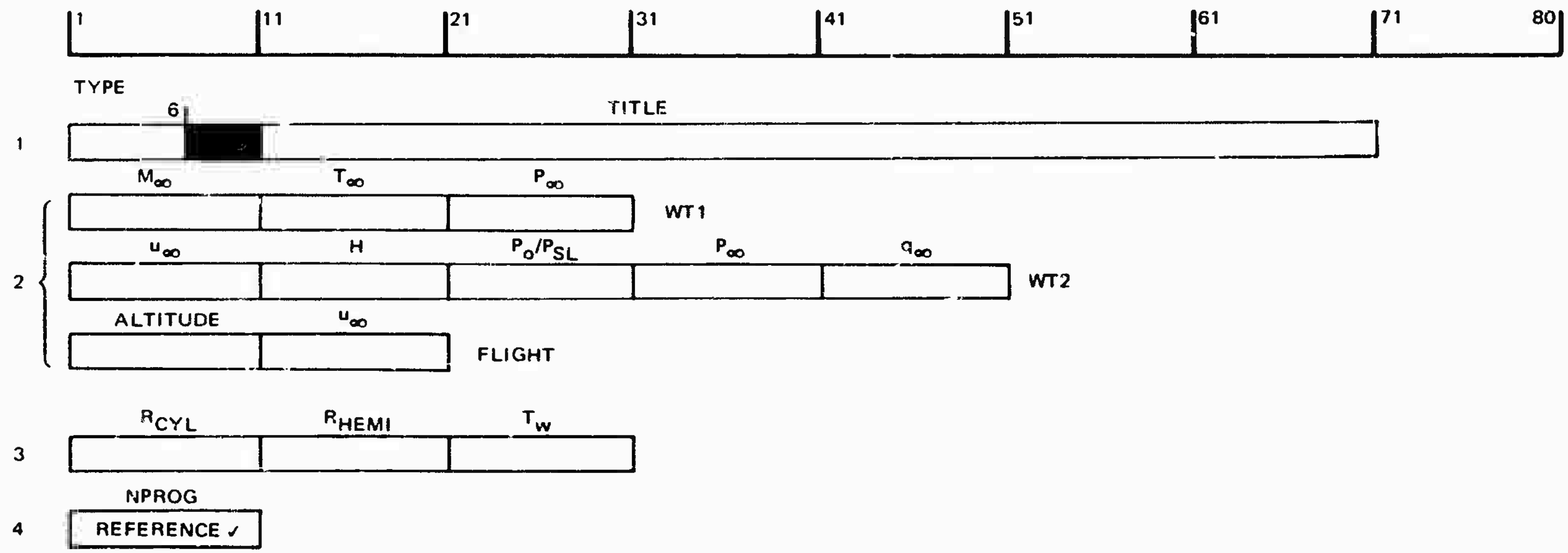

1
11
21
31
41
51
61
71
80
9
η_{max} TABLE (M VALUES)
10
θ_{max} TABLE (M VALUES)
11
ϑ_e TABLE (M x N VALUES)
12
P/P_o TABLE (M x N VALUES)
13
MT
NT
5
10



1
11
21
31
41
51
61
71
80
ξ TABLE (MT VALUES)
14
η TABLE (NT VALUES)
15
T_w TABLE (MT x NT VALUES)
16
K_{PO}
5
$R_{r,transition}$
$\Delta\eta_f$
17
x TABLE (K_{PO} VALUES)
18

6) REFERENCE CALCULATIONS (PROGRAM REF)

Card	Columns	Format	Description
1	Same as Option D-1		
2			
3			
4	1-10	A6	Must contain the word REFERENCE.

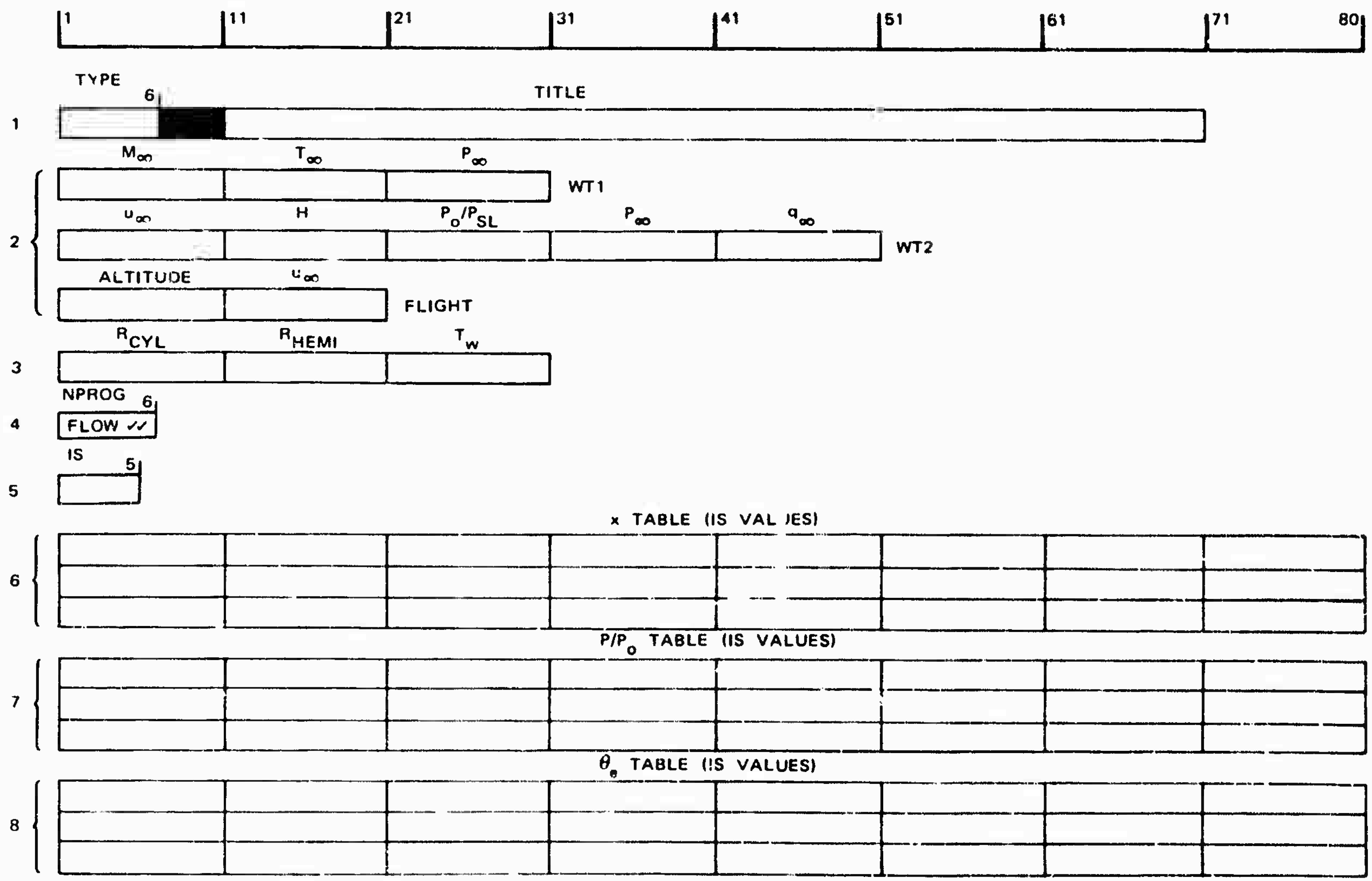


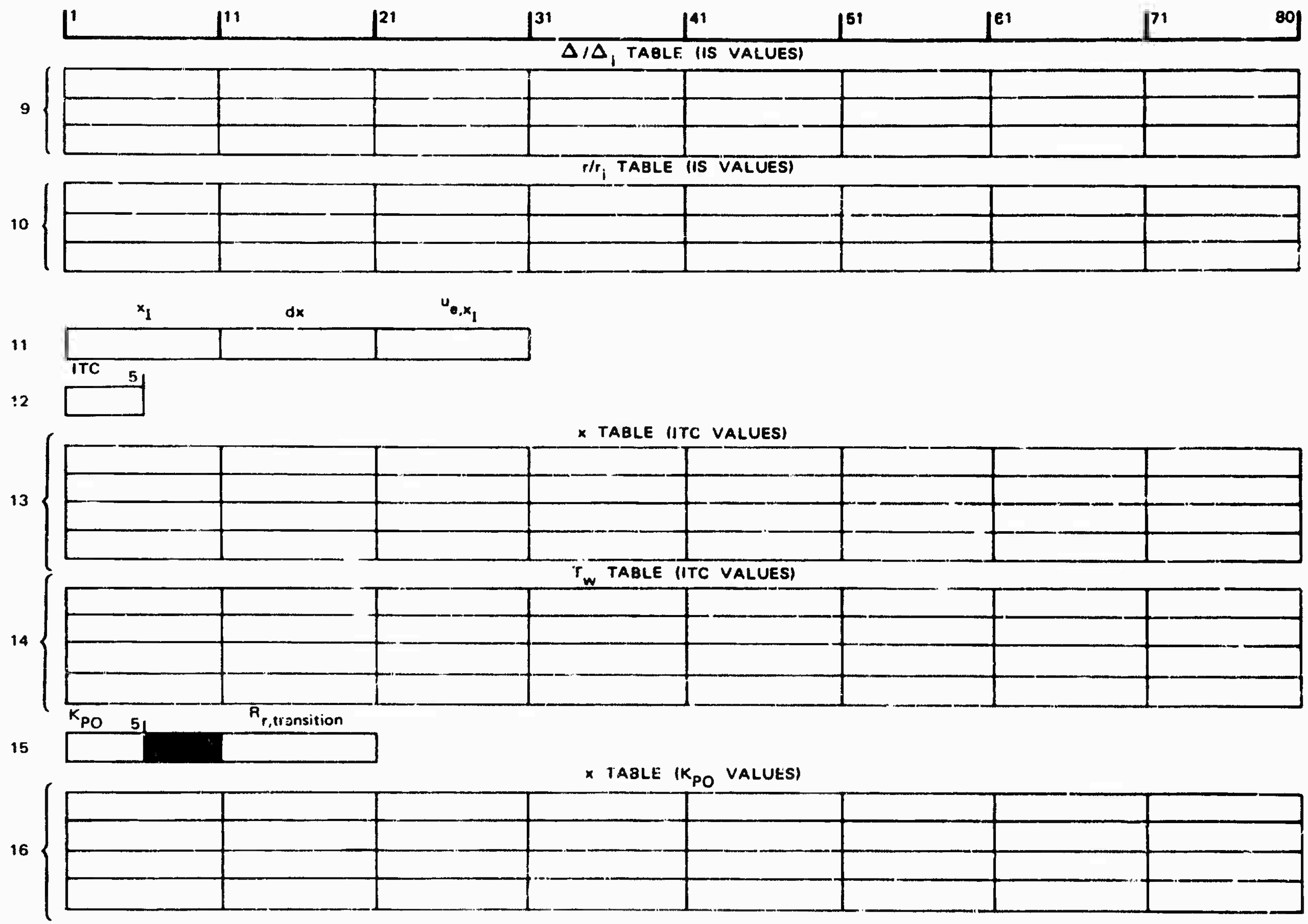
1
11
21
31
41
51
61
71
80
TYPE
6
TITLE
1
M_∞
T_∞
P_∞
WT1
u_∞
H
P_o/P_{SL}
P_∞
q_∞
2
WT2
ALTITUDE
u_∞
FLIGHT
R_{CYL}
R_{HEMI}
T_w
3
NPROG
4
REFERENCE ✓

7) BOUNDARY LAYER CALCULATIONS (PROGRAM B)

Card	Columns	Format	Description
1 2 3	Same as Option D-1		
4	1-6	A6	Must contain the word FLOW.
5	1-5	I5	IS is the number of values in the P/P_o, θ_e, Δ/Δ_i and r/r_i tables as a function of streamline distance. $2 \leq IS \leq 50$.
6	----	8F10	This is a table containing IS number of x locations, ft, along the streamline at which P/P_o, θ_e, Δ/Δ_i and r/r_i will be input.
7	----	8F10	This is a table containing IS values of local pressure each normalized by the model or vehicle stagnation pressure. The P/P_o correspond to the above x table.
8	----	8F10	This table contains IS values of the local streamline angle relative to the axis of symmetry, degrees. The values of θ_c correspond to the above x table, degrees.
9	----	8F10	This table contains IS values of the streamline divergence parameter Δ/Δ_i. The values of Δ/Δ_i correspond to the above x table.
10	---	8F10	This table contains IS values of the body shock geometry parameter r/r_i. The values of r/r_i correspond to the above x table.
11	1-10	F10	x_I is point at which the calculation will begin, ft.

Card	Column	Format	Description
11	11-20	F10	dx is the increment in distance along the streamline. ft $$dx > \frac{x_{FINAL} - x_I}{750}$$
11	21-30	F10	u_{e,x_I} is the edge velocity at ft/sec.
12	1-5	I5	ITC is the number of values in the T_w versus x table. $2 \leq ITC \leq 50$. If T_w = constant then ITC = 2.
13	----	8F10	This is a table of x locations (ITC) along the streamline where T_w will be input.
14	----	8F10	This is a table of ITC values of T_w along the streamline corresponding to the x table. If ITC = 2 enter only two values of TW, °R.
15	1-5	I5	K_{PO} is the number of x locations along the streamline at which printout is desired. $K_{PO} < 500$.
15	11-20	F10	Transition Reynolds number. see definition of $R_{r.Q}$ in Program A. labeled QTRANS in section 2. b. 18.
16	----	8F10	This is a table of K_{PO} locations along the streamline where printout is desired.

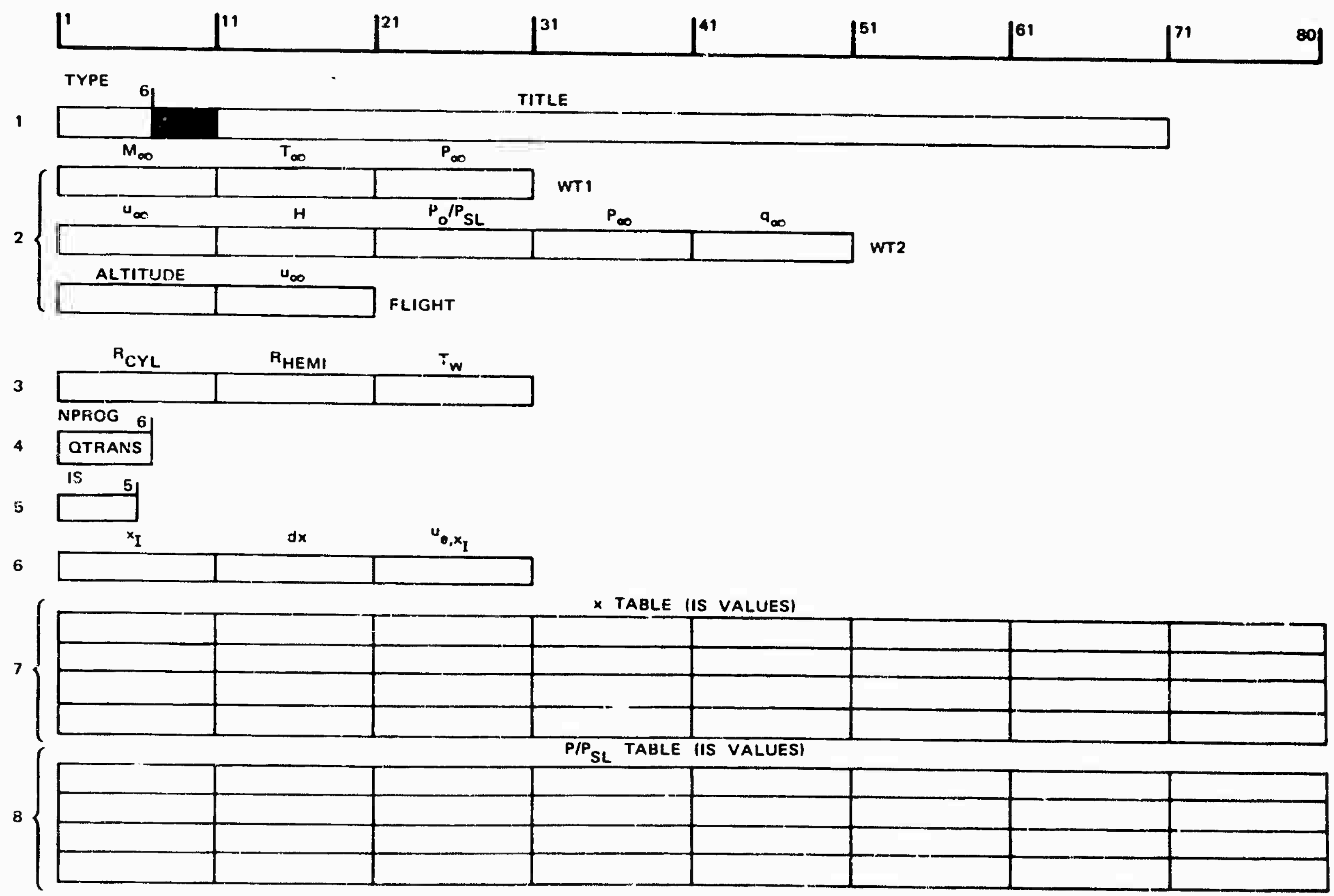

1
11
21
31
41
51
61
71
80
TYPE
6
TITLE
1
M_∞
T_∞
P_∞
WT1
u_∞
H
P_o/P_{SL}
P_∞
q_∞
WT2
2
ALTITUDE
u_∞
FLIGHT
R_{CYL}
R_{HEMI}
T_w
3
NPROG
6
4
FLOW ✓✓
IS
5
5
x TABLE (IS VAL JES)
6
P/P_o TABLE (IS VALUES)
7
θ_e TABLE (IS VALUES)
8


1
11
21
31
41
51
61
71
80
9
Δ/Δ_i TABLE (IS VALUES)
10
r/r_i TABLE (IS VALUES)
11
x_I
dx
u_{e,x_I}
12
ITC
5
13
x TABLE (ITC VALUES)
14
T_w TABLE (ITC VALUES)
15
K_{PO}
5
$R_{r,transition}$
16
x TABLE (K_{PO} VALUES)

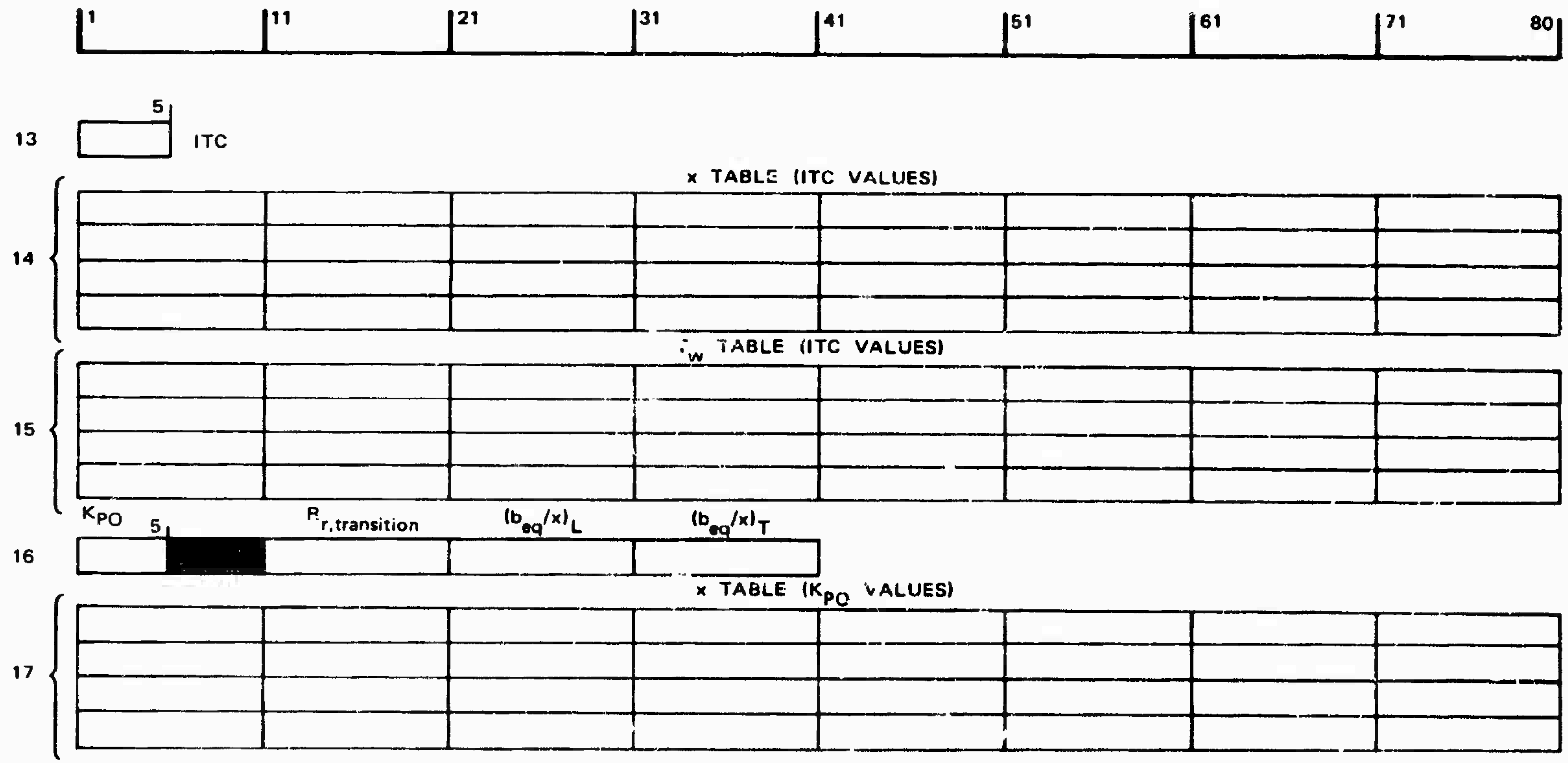
8) HEAT TRANSFER CALCULATIONS (PROGRAM A)

Card	Columns	Format	Description
1	Same as Option D-1		
2			
3			
4	1-6	A6	Must contain the word QTRANS.
5	1-5	I5	IS is the number of values in each of the following tables: x. P/P_{SL}, u_e, θ_e, Δ/Δ_i and r/r_i. $2 \leq IS \leq 50$.
6	1-10	F10	x_I is the initial x location along the streamline at which the calculations will start, ft.
6	11-20	F10	dx is the increment in distance along the streamline. (ft.) $dx > \frac{x_{FINAL} - x_I}{750}$
6	21-30	F10	$u_{e.x_I}$ is the boundary layer edge velocity at x_I, ft/sec.
7	----	8F10	This is a table containing x locations (IS) along the streamline at which P/P_{SL}, u_e, θ_e, Δ/Δ_i and r/r_i will be specified, ft.
8	----	8F10	This is a table containing IS values of the local pressure normalized with the sea level pressure. and at locations corresponding to the x table.
9	----	8F10	This is a table containing IS values of the boundary layer edge velocity corresponding to locations specified in the x table. ft/sec.
10	----	8F10	This is a table containing IS values of the local streamline angle θ_e relative to the axis of symmetry. The angles correspond to x locations specified in the x table. deg.

Card	Columns	Format	Description
11	----	8F10	This is a table containing IS values of the streamline divergence parameter Δ/Δ_i and correspond to x locations specified in the x table.
12	----	8F10	This is a table containing IS values of the body-shock parameter r/r_i and correspond to x locations specified in the x table.
13	1-5	I5	ITC is the number of values in the T_w versus x table. $2 \leq ITC \leq 50$ If T_w is a constant, ITC = 2. Input two values of T_w into card 15.
14	----	8F10	This is a table containing ITC values of x along the streamline, ft.
15	----	8F10	This is a table of ITC values of T_w along the streamline and corresponding to the x locations in the preceding table. (°R)
16	1-5	I5	K_{PO} is the number of x printout locations desired along the streamline. $K_{PO} < 500$.
16	11-20	F10	Transition Reynolds number, see definition of $R_{r,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
16	21-30	F10	BEQL is the ratio of the laminar equivalent distance parameter to the distance along the streamline. If $x_I > 0$ then a value of BEQL is required. If $x_I = 0$ input BEQL=0.
16	31-40	F10	BEQT is the turbulent equivalent distance parameter. See BEQL above.
17	----	8F10	This is a table of K_{PO} locations along the streamline where printout is desired.


1
11
21
31
41
51
61
71
80
TYPE
6
TITLE
1
M_∞
T_∞
P_∞
WT1
u_∞
H
P_o/P_{SL}
P_∞
q_∞
WT2
2
ALTITUDE
u_∞
FLIGHT
R_{CYL}
R_{HEMI}
T_w
3
NPROG
6
QTRANS
4
IS
5
5
x_I
dx
u_{e,x_I}
6
x TABLE (IS VALUES)
7
P/P_{SL} TABLE (IS VALUES)
8

Card	1	11	21	31	41	51	61	71	80
	u_e TABLE (IS VALUES)								
9									
	θ_e TABLE (IS VALUES)								
10									
	Δ/Δ_i TABLE (IS VALUES)								
11									
	r/r_i TABLE (IS VALUES)								
12									


1
11
21
31
41
51
61
71
80
13
5
ITC
14
x TABLE (ITC VALUES)
15
x_w TABLE (ITC VALUES)
16
K_{PO}
5
$R_{r,transition}$
$(b_{eq}/x)_L$
$(b_{eq}/x)_T$
17
x TABLE (K_{PO} VALUES)

b. Output Description

INITIAL CONDITIONS

VELOCITY	free stream velocity, ft/sec
MACH	free stream Mach number
ENTHALPY	free stream enthalpy, ft^2/sec^2
TINF	free stream static temperature, °R
PINF	free stream static pressure, lb/ft^2
QINF	free stream dynamic pressure, lb/ft^2
PO/PSL	(model total pressure)/(sea level static pressure)
PO/PINF	(model total pressure)/(free stream static pressure)

REFERENCE CONDITIONS

HO	$\dot{q}_o/(i_{aw}-i_w) = h_o/C_P$ where h_o is the hemisphere stagnation point heat transfer coefficient, lb_m/ft^2-sec
.24HO	$.24\ h_o/C_P$, Btu/ft^2-sec-° F.
QDOTO	hemisphere stagnation point heating rate, Btu/ft^2-sec
HREF, L	$\dot{q}_{REF,L}/(i_{aw,L} - i_{w,REF}) = h_{REF,L}/C_P$ where $h_{REF,L}$ is the laminar heat transfer coefficient based on enthalpy at the stagnation line of a 60° swept infinite cylinder, lb_m/ft^2-sec
RHEM, REF	hemisphere radius, ft
RCYL, REF	cylinder radius, ft
SWEEP	sweep angle of the swept cylinder used for the turbulent reference condition, degrees
.24HREF, L	$.24\ h_{REF,L}/C_P$
QDOT, REF, L	laminar heating rate at the stagnation line of a 60° swept infinite cylinder, Btu/ft^2-sec

HREF,T	$\dot{q}_{REF,T}/(i_{aw,T} - i_{w,REF}) = h_{REF,T}/C_P$ where $h_{REF,T}$ is the turbulent heat transfer coefficient at the stagnation line of a 60° swept infinite cylinder based on enthalpy, lb_m/ft^2-sec
.24HREF.T	$.24\ h_{REF,T}/C_P$
QDOT,REF,T	turbulent heating rate at the stagnation line of a 60° swept infinite cylinder, Btu/ft^2-sec-°R
IAW,REF,L	laminar adiabatic wall enthalpy on a 60° swept infinite cylinder, ft^2/sec^2
IAW,REF,T	turbulent adiabatic wall enthalpy on a 60° swept infinite cylinder, ft^2/sec^2
IW,REF	reference wall enthalpy, ft^2/sec^2
TAUREF,L	laminar shear stress on a 60° swept infinite cylinder, lb_f/ft^2
CFINF,REF,L	$2\tau_{ref,L}/(\rho_\infty u_\infty^2)$, laminar skin friction coefficient based on free stream conditions
IW,O	stagnation wall enthalpy, ft^2/sec^2
TAUREF,T	turbulent shear stress on a 60° swept infinite cylinder, lb_f/ft^2
CFINF,REF,T	$2\tau_{ref,T}/(\rho_\infty u_\infty^2)$, turbulent skin friction coefficient based on free stream conditions
TW,REF	wall temperature for 60° swept infinite cylinder heating calculations, °R

GEOMETRIC PARAMETERS

SWEEP CYL	cylinder sweep angle, degrees
SWEEP DELTA WING	delta wing sweep angle, degrees
ALPHA	angle of attack, degrees
RCYL	cylinder radius, ft
RHEMI	hemisphere radius, ft

STREAMLINE

XSI	coordinate, ft
ETA	coordinate, ft
THETAE	streamline angle, degrees

BOUNDARY LAYER CALCULATIONS

X	streamline distance, ft
ETA	ordinate of coordinate system, ft
XSI	abscissa of coordinate system, ft
BEQXL	$(b_{eq}/x)_L$, laminar equivalent distance parameter
BEQXT	$(b_{eq}/x)_T$, turbulent equivalent distance parameter
POPSL	local pressure in atmospheres
AYEAWL	$i_{aw,L}$, adiabatic wall enthalpy in laminar flow, ft^2/sec^2
AYEAWT	$i_{aw,T}$, adiabatic wall enthalpy in turbulent flow, ft^2/sec^2
UE	edge velocity, ft/sec
HL	laminar heat transfer coefficient divided by the reference hemisphere stagnation point heat transfer coefficient
HT	turbulent heat transfer coefficient divided by the reference turbulent heat transfer coefficient at the stagnation line of a 60° swept infinite cylinder
DODI	Δ/Δ_i, total streamline divergence
N	$x/(\Delta/\Delta_i)\,[d(\Delta/\Delta_i)/dx]$, streamline divergence parameter
QLISO	laminar isothermal heat transfer rate, Btu/ft^2-sec-°R
QTISO	turbulent isothermal heat transfer rate, Btu/ft^2-sec-°R
RORI	r/r_i, streamline divergence due to body-shock geometry

QDOTL	laminar heating rate, Btu/ft^2-sec-°R
QDOTT	turbulent heating rate, Btu/ft^2-sec-°R
TW	wall temperature, °R
CFEL	laminar skin friction coefficient (based on dynamic pressure at boundary layer edge)
CFET	turbulent skin friction coefficient (based on dynamic pressure at boundary layer edge)
ZTEX	ZT_e, compressibility factor times the edge temperature, °R
RRQ	heat transfer reference Reynolds number
RRS	skin friction reference Reynolds number
OMEGAE	edge viscosity divided by ZT_e, lb_f-sec/ft^2-°R
AIEE	i_e, edge enthalpy, ft^2/sec^2
AYEW	i_w, wall enthalpy, ft^2/sec^2
THETA	θ_e, local streamline angle, degrees
MUO	μ_o, absolute viscosity at the stagnation reference condition, lb_f-sec/ft^2
ZTR	compressibility factor-temperature product evaluated at the reference enthalpy, °R
EBARL	laminar crossflow pressure gradient parameter
EBART	turbulent crossflow pressure gradient parameter
JL	laminar streamwise pressure gradient profile parameter
XEQL	laminar heat transfer equivalent distance divided by the local distance along the streamline
FXQ	see definition of $F_{x,Q}$ in QTRANS
FXS	see definition of $F_{X,S}$ in QTRANS
OMEGAR	reference viscosity divided by reference temperature, lb_f-sec/ft^2-°R

THETAL	laminar momentum thickness, ft
THETAT	turbulent momentum thickness, ft

c. Error Statements

ERROR IN CONTROL CARD (NTYPE = name)

ERROR IN CONTROL CARD (NPROG = name)

The above errors are caused by an input error of the code words NTYPE or NPROG. "Name" is the control word that was input. This error will terminate the run.

ALTITUDE INPUT TO ATMOS EXCEEDED RANGE - ASSIGN VALUE OF 2,300,000 FT.

ALTITUDE INPUT TO ATMOS BELOW ROUTINE RANGE - ASSIGN VALUE OF 0 FT.

Input altitude out of range of ATMOS subroutine. FLIGHT case only.

STREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SUBROUTINE AXI2D

The input value of dx is too small for an axisymmetric or two-dimensional case. Also printed with this comment is the x-value at which the array dimension was exceeded, the maximum x defined by the geometry input, and the input dx. The run is terminated by this error.

DBTP ERROR

The independent variable is outside the range of the program tables. The independent variable and the table limits are printed. The program will use the table limit and proceed.

STREAMLINE CROSS AT XSI =

When the two divergence calculation streamlines cross, the above comment is printed with the value of the ξ coordinate at that point. The program moves downstream, by dx increments, until the streamlines are not crossed and begins calculations at that point.

CONVERGENCE NOT ESTABLISHED IN VELOCITY CALCULATION

The velocity iteration routine in subroutine FLOW failed to converge on a solution in 10 iterations. The program proceeds using the final calculated velocity.

SIRCH ERROR

Table interpolation error in the general interpolation routine SIRCH.

ERROR IN TABLE2 LOOK-UP

ERROR IN TABL18 LOOK-UP

An error occurred during table interpolation in subroutine TABLE2 or TABL18.

FREE STREAM ENTHALPY EXCEEDS TABLE VALUES IN SONENT

The enthalpy value used for table interpolation in subroutine SONENT is outside the table limits.

ERROR IN STREAMLINE CALCULATION

This comment is printed when a calculated streamline point lies outside the $\theta_e(\xi, \eta)$ grid in subroutine STREAM.

4. PROGRAMMING INFORMATION

a. $\rho_r \mu_r$ Program Listing

```
$IBFTC AS2419  DECK                                                     24190001
C        MAIN PROGRAM FOR RHO-MU PROGRAM (AS2419)                       24190002
      COMMON/Q/A(11600),IA(10)                                          24190003
      EQUIVALENCE                                                       24190004
     1(A(    1),ACH    ) , (A(    2),ALPHA ) , (A(    3),ALT    ),      24190005
     2(A(    4),AYEWO  ) , (A(    5),BEQXL ) , (A(    6),BEQXT  ),      24190006
     3(A(    7),DELETA) , (A(    8),DELXSI) , (A(    9),DETA   ),       24190007
     4(A(   10),DETAS  ) , (A(   11),DX    ) , (A(   12),DXP    ),      24190008
     5(A(   13),ETAF   ) , (A(   14),ETAI  ) , (A(   15),H      ),      24190009
     6(A(   16),HO     ) , (A(   17),HREFT ) , (A(   18),PINF   ),      24190010
     7(A(   19),POPINF) , (A(   20),POPSL ) , (A(   21),QINF   ),       24190011
     8(A(   22),RADIUS) , (A(   23),RCLREF) , (A(   24),RHMREF),        24190012
     9(A(   25),RHO    ) , (A(   26),RTRANS) , (A(   27),SWEEP  ),      24190013
     1(A(   28),TEMP   ) , (A(   29),TWREF ) , (A(   30),URATIO),       24190014
     2(A(   31),VEL    ) , (A(   32),VELP  ) , (A(   33),XF     ),      24190015
     3(A(   34),XI     ) , (A(   35),XSIF  ) , (A(   36),XSII   ),      24190016
     4(A(   37),XSI    ) , (A(   87),ETA   ) , (A(   87),Y      ),      24190017
     5(A(  137),ETAMAX) , (A(  187),TH8MX ) , (A(  237),THETAE),        24190018
     6(A(  237),POPSLX) , (A(  987),DODI  ) , (A( 1737),RORI   ),       24190019
     7(A( 2737),PRESSR) , (A( 2737),AIEE  ) , (A( 3487),OMEGAE),        24190020
     8(A( 4237),ZTEX   ) , (A( 5237),TWALL ) , (A( 5237),X      ),      24190021
     9(A( 5987),XSIX   ) , (A( 6737),ETAX  ) , (A( 7737),XSTAR )        24190022
      EQUIVALENCE                                                       24190023
     1(A( 8487),PRESRX) , (A( 9237),THETAS) , (A( 9987),UE     ),       24190024
     2(A(10737),TW     ) , (A(11487),XPO   ) , (A(11587),P      )       24190025
      EQUIVALENCE                                                       24190026
     1(IA(    1),M      ) , (IA(    2),N     ) , (IA(    3),KF     ),   24190027
     3(IA(    4),II     ) , (IA(    8),NPR   ) , (IA(    9),ITC    )    24190028
      DIMENSION XSI(50),ETA(50),Y(50),ETAMAX(50),TH8MX(50),             24190029
     1THETAE(50,50),PRESSR(50,50),XSTAR(750),PRESRX(750),POPSLX(750),   24190030
     2UE(700),AIEE(750),ZTEX(750),OMEGAE(750),THETAS(750),DODI(750),    24190031
     3 RORI(750),TW(750),XPO(100),TWALL(50,50)                          24190032
      DIMENSION P(5),ETAX(750),XSIX(750),X(750)                         24190033
      DIMENSION DODS(50),IPROG(9),ITYPE(3),PS(50),RORS(50),TWS(50),     24190034
     1UES(50),XS(50),THETS(50)                                          24190035
      DIMENSION XSIT(50),ETAT(50)                                       24190036
      DIMENSION TITLE(10)                                               24190037
      DATA (ITYPE(I),I=1,3)/6HWT1    ,6HWT2    ,6HFLIGHT/               24190038
      DATA(IPROG(I),I=1,9)/6HREFERE,6HAXISYM,6HTWODIM,6HHEMISP,         24190039
     16HDELTA ,6HCYLIND,6HSTREAM,6HFLOW   ,6HQTRANS/                    24190040
 5000 FORMAT(15I5)                                                      24190041
```

```
 5010 FORMAT(8F10.0)
 5020 FORMAT(A6,4X,10A6)
 5050 FORMAT(I5,5X,5F10.0)
 5060 FORMAT(2F10.0,10I5)
 1000 CONTINUE
      IERROR=0
      NXSV=1
      READ(5,5020)NTYPE,(TITLE(I),I=1,10)
      WRITE(6,6001)(TITLE(I),I=1,10)
 6001 FORMAT(1H1,10A6)
      DO 1 I=1,3
      L1=I
      IF(NTYPE-ITYPE(I))1,3,1
    1 CONTINUE
      WRITE(6,6010)NTYPE
 6010 FORMAT(1H0,29HERROR IN CONTROL CARD (NTYPE=,A6,2H ))
      STOP
    3 CONTINUE
C
      GO TO (10,20,30),L1
   10 CONTINUE
      READ(5,5010)ACH,TEMP,PINF
      GO TO 40
   20 CONTINUE
      READ(5,5010)VEL,H,POPSL,PINF
      GO TO 40
   30 CONTINUE
      READ(5,5010)ALT,VEL
   40 CONTINUE
      READ(5,5010)RCLREF,RHMREF,TWREF
      READ(5,5020)NPROG
      WRITE(6,6015)NPROG
 6015 FORMAT(1H0,A6)
      CALL INTIAL(L1,NTYPE)
C
      CALL REF
      DO 60 I=1,9
      IF(NPROG-IPROG(I))60,50,60
   50 NPR=I
      GO TO 70
   60 CONTINUE
      WRITE(6,6020)NPROG
```

```
 6020 FORMAT(30H0ERROR IN CONTROL CARD (NPROG=,A6,1H))
      STOP
C
   70 CONTINUE
      GO TO(100,200,200,400,500,300,600,700,800),NPR
C
  100 CONTINUE
      GO TO 1000
C
  200 CONTINUE
C        AXISYMMETRIC AND TWO-DIMENSIONAL
      READ(5,5000)M
      READ(5,5010)(XSI(I),I=1,M)
      READ(5,5010)(Y(I),I=1,M)
      READ(5,5010)XI,DX
      CALL AXI2D
      READ(5,5000)ITC
      READ(5,5010)(XS(I),I=1,ITC)
      READ(5,5010)(TWS(I),I=1,ITC)
      READ(5,5050)KF,RTRANS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL WALLT1(XS,TWS)
      CALL BEQI
      CALL QTRAN
      GO TO 1000
C
  300 CONTINUE
C        SWEPT CYLINDER
      READ(5,5010)RADIUS,SWEEP
      READ(5,5010)XSIF,ETAF,DX
      WRITE(6,6300) RADIUS,SWEEP,XSIF,ETAF,DX
 6300 FORMAT(1H0,14HSWEPT CYLINDER //3X,6HRADIUS,6X,5HSWEEP,7X,4HXSIF,8X
     1,4HETAF,8X,2HDX//5E12.5)
      READ(5,5000)N
      ITC=N
      READ(5,5010)(ETAT(I),I=1,ITC)
      READ(5,5010)(TWS(I),I=1,ITC)
      READ(5,5050)KF,RTRANS,DETAS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL CYLIND
      CALL STREAM
```

```
      CALL DIVERG                                                       24190125
      CALL FLOW                                                         24190126
      NXT=1                                                             24190127
      DO 350 I=1,II                                                     24190128
  350 TW(I)=TBLP(ETAT,TWS,ETAX(I),ITC,NXT)                              24190129
      CALL BEQI                                                         24190130
      CALL QTRAN                                                        24190131
      GO TO 1000                                                        24190132
C                                                                       24190133
  400 CONTINUE                                                          24190134
C        HEMISPHERE                                                     24190135
      READ(5,5010)RADIUS                                                24190136
      WRITE(6,6400)RADIUS                                               24190137
 6400 FORMAT(1H0,19HHEMISPHERE RADIUS =  ,E12.5)                        24190138
      CALL HEMI                                                         24190139
      CALL FLOW                                                         24190140
  410 CONTINUE                                                          24190141
      READ(5,5000)ITC                                                   24190142
      READ(5,5010)(XS(I),I=1,ITC)                                       24190143
      READ(5,5010)(TWS(I),I=1,ITC)                                      24190144
      READ(5,5050)KF,RTRANS                                             24190145
      READ(5,5010)(XPO(I),I=1,KF)                                       24190146
      CALL WALLT1(XS,TWS)                                               24190147
      CALL BEQI                                                         24190148
      CALL QTRAN                                                        24190149
      GO TO 1000                                                        24190150
C                                                                       24190151
  500 CONTINUE                                                          24190152
C        DELTA WING                                                     24190153
      READ(5,5000)ITC                                                   24190154
      XSII=0.                                                           24190155
      ETAI=0.                                                           24190156
      READ(5,5010)ALPHA,SWEEP,XSIF,ETAF,XI                              24190157
      READ(5,5010)DELXSI,DELETA,DX                                      24190158
      WRITE(6,6500)ALPHA,SWEEP,XSII,ETAI,XSIF,ETAF,DELXSI,              24190159
     1DELETA,DX                                                         24190160
 6500 FORMAT(11H0DELTA WING //3X,5HALPHA,7X,5HSWEEP,7X,4HXSII,8X,4HETAI,24190161
     18X,4HXSIF,8X,4HETAF,8X,6HDELXSI,6X,6HDELETA,6X,2HDX//9E12.5)      24190162
      CALL DELTA                                                        24190163
      CALL STREAM                                                       24190164
      IF(ITC .EQ. 1)GO TO 510                                           24190165
      READ(5,5000)MT,NT                                                 24190166
```

```
      READ(5,5010)(XSIT(I),I=1,MT)
      READ(5,5010)(ETAT(I),I=1,NT)
      READ(5,5010)((TWALL(I,J),I=1,MT),J=1,NT)
      GO TO 520
  510 CONTINUE
      READ(5,5010)TWC
      DO 515 I=1,II
      TW(I)=TWC
  515 CONTINUE
  520 CONTINUE
      READ(5,5050)KF,RTRANS,DETAS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL DIVERG
      IF(ITC .EQ. 1) GO TO 505
      CALL WALLT2(MT,NT,XSIT,ETAT)
  505 CONTINUE
      CALL FLOW
      CALL BEQI
      CALL QTRAN
      GO TO 1000
C
  600 CONTINUE
C        STREAMLINE
      READ(5,5000)M,N,ITC
      READ(5,5010)XSIF,ETAF,DX,XI,URATIO
      READ(5,5010)(XSI(I),I=1,M)
      READ(5,5010)(ETA(J),J=1,N)
      READ(5,5010)(ETAMAX(J),J=1,M)
      READ(5,5010)(TH8MX(J),J=1,M)
      READ(5,5010)((THETAE(I,J),I=1,M),J=1,N)
      READ(5,5010)((PRESSR(I,J),I=1,M),J=1,N)
      URATIO=URATIO/VEL
      IF(ITC .EQ. 1)GO TO 610
      READ(5,5000)MT,NT
      READ(5,5010)(XSIT(I),I=1,MT)
      READ(5,5010)(ETAT(I),I=1,NT)
      READ(5,5010)((TWALL(I,J),I=1,MT),J=1,NT)
      GO TO 620
  610 CONTINUE
      READ(5,5010)TWC
  620 CONTINUE
```

```
      READ(5,5050)KF,RTRANS,DETAS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL STREAM
      CALL DIVERG
      IF(ITC .EQ. 1) GO TO 625
      CALL WALLT2(MT,NT,XSIT,ETAT)
      GO TO 630
  625 DO 615 I=1,II
  615 TW(I)=TWC
  630 CONTINUE
      CALL FLOW
      CALL BEQI
      CALL QTRAN
      GO TO 1000
C
  700 CONTINUE
C        FLOW CONDITIONS
      READ(5,5000)IS
      READ(5,5010)(XS(I),I=1,IS)
      READ(5,5010)(PS(I),I=1,IS)
      READ(5,5010)(THETS(I),I=1,IS)
      READ(5,5010)(DODS(I),I=1,IS)
      READ(5,5010)(RORS(I),I=1,IS)
      READ(5,5010)XI,DX,URATIO
      II=(XS(IS)-XI)/DX
      II=II+1
      XSTAR(1)=XI
      DO 720 I=1,II
      XSTAR(I+1)=XSTAR(I)+DX
      X(I)=XSTAR(I)
      XSIX(I)=0.
      ETAX(I)=0.
      PRESRX(I)=TBLP(XS,PS,XSTAR(I),IS,NXSV)
      THETAS(I)=TBLP(XS,THETS ,XSTAR(I),IS,NXSV)
      DODI(I)=TBLP(XS,DODS,XSTAR(I),IS,NXSV)
      RORI(I)=TBLP(XS,RORS,XSTAR(I),IS,NXSV)
  720 CONTINUE
      P(1)=PS(1)
      P(2)=PS(2)
      P(3)=PS(3)
      DXP=.5*(XS(3)-XS(1))
      URATIO=URATIO/VEL
```

```
      READ(5,5000)ITC
      READ(5,5010)(XS(I),I=1,ITC)
      READ(5,5010)(TWS(I),I=1,ITC)
      CALL WALLT1(XS,TWS)
      READ(5,5050)KF,RTRANS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL FLOW
      CALL BEQI
      CALL QTRAN
      GO TO 1000
C
  800 CONTINUE
C        QTRANSFER
      READ(5,5000)IS
      READ(5,5010)XI,DX,URATIO
      READ(5,5010)(XS(I),I=1,IS)
      READ(5,5010)(PS(I),I=1,IS)
      READ(5,5010)(UES(I),I=1,IS)
      READ(5,5010)(THETS(I),I=1,IS)
      READ(5,5010)(DODS(I),I=1,IS)
      READ(5,5010)(RORS(I),I=1,IS)
      II=(XS(IS)-XS(1))/DX
      II=II+1
      XSTAR(1)=XS(1)
      X(1)=XSTAR(1)
      XSIX(1)=0.
      ETAX(1)=0.
      POPSLX(1)=PS(1)
      UE(1)=UES(1)
      AIEE(1)=H-UE(1)**2/2.
      CALL SOMEGA(AIEE(1),POPSLX(1),ZTEX(1),OMEGAE(1))
      THETAS(1)=THETS(1)
      DODI(1)=DODS(1)
      RORI(1)=RORS(1)
      DO 820 I=2,II
      XSTAR(I)=XSTAR(I-1)+DX
      X(I)=XSTAR(I)
      XSIX(I)=0.
      ETAX(I)=0.
      POPSLX(I)=TBLP(XS,PS,XSTAR(I),IS,NXSV)
      UE(I)=TBLP(XS,UES,XSTAR(I),IS,NXSV)
```

```
      AIEE(I)=H-UE(I)**2/2.                                             24190291
      CALL SOMEGA(AIEE(I),POPSLX(I),ZTEX(I),OMEGAE(I))                  24190292
      THETAS(I)=TBLP(XS,THETS,XSTAR(I),IS,NXSV)                         24190293
      DODI(I)=TBLP(XS,DODS,XSTAR(I),IS,NXSV)                            24190294
      RORI(I)=TBLP(XS,RORS,XSTAR(I),IS,NXSV)                            24190295
  820 CONTINUE                                                          24190296
      URATIO=URATIO/VEL                                                 24190297
      READ(5,5000)ITC                                                   24190298
      READ(5,5010)(XS(I),I=1,ITC)                                       24190299
      READ(5,5010)(TWS(I),I=1,ITC)                                      24190300
      CALL WALLT1(XS,TWS)                                               24190301
      READ(5,5050)KF,RTRANS,BEQXL,BEQXT                                 24190302
      READ(5,5010)(XPO(I),I=1,KF)                                       24190303
      IF(BEQXL)825,825,830                                              24190304
  825 CONTINUE                                                          24190305
      CALL BEQI                                                         24190306
  830 CONTINUE                                                          24190307
      CALL QTRAN                                                        24190308
      GO TO 1000                                                        24190309
      END                                                               24190310
$IBFTC DBTP     DECK                                                    24190311
C     FUNCTION DBTP USES TBLP TO PERFORM DOUBLE LINEAR INTERPOLATION    24190312
      FUNCTION DBTP(XX,YY,ZZ,X,Y,NX,NY,NXS,NYS)                         24190313
      DIMENSION XX(NX),YY(NY),ZZ(50,50)                                 24190314
C                                                                       24190315
      ML=0                                                              24190316
      IF(X-XX(1))10,20,30                                               24190317
   10 ML=1                                                              24190318
   20 NXS=1                                                             24190319
      X1=XX(1)                                                          24190320
      X2=0.                                                             24190321
      GO TO 170                                                         24190322
   30 CONTINUE                                                          24190323
      IF(X-XX(NX))60,50,40                                              24190324
   40 ML=1                                                              24190325
   50 CONTINUE                                                          24190326
      NXS=NX                                                            24190327
      X1=XX(NX)                                                         24190328
      X2=0.                                                             24190329
      GO TO 170                                                         24190330
   60 CONTINUE                                                          24190331
      IF(X-XX(NXS))80,70,120                                            24190332
```

```
   70 X1=XX(NXS)
      X2=0.
      GO TO 170
   80 N=NXS-1
   90 CONTINUE
      IF(X-XX(N))100,160,110
  100 N=N-1
      GO TO 90
  110 X1=XX(N)
      X2=XX(N+1)
      NXS=N
      GO TO 170
  120 CONTINUE
      N=NXS
  130 CONTINUE
      IF(X-XX(N))150,160,140
  140 N=N+1
      GO TO 130
  150 X2=XX(N)
      X1=XX(N-1)
      NXS=N-1
      GO TO 170
  160 X1=XX(N)
      X2=0.
      NXS=N
  170 CONTINUE
      IF(ML)172,175,172
  172 WRITE(6,6010)N,X,XX(1),NX,XX(NX)
 6010 FORMAT(1H0,10HDBTP ERROR/5H N= ,I3,5H X = ,E12.5,8H X(1) = ,E12.5,
     13H X(,I3,4H) = ,E12.5)
  175 CONTINUE
      W=TBLP(YY,ZZ(1,NXS),Y,NY,NYS)
      IF(X2)190,190,180
  180 R=TBLP(YY,ZZ(1,NXS+1),Y,NY,NYS)
      Z=R-(R-W)*(X2-X)/(X2-X1)
      GO TO 200
  190 Z =W
  200 DBTP=Z
      RETURN
      END
$IBFTCTTBLP    DECK
```

```
      FUNCTION TBLP(XT,YT,X,NTAB,N)
      DIMENSION XT(NTAB),YT(NTAB)
C        THIS IS A MODIFICATION OF THE UNIVAC 1107 ROUTINE
C
      DELY=0.
      M=0
      IF(X-XT(1))1,2,3
    1 M=-1
    2 N=1
      GO TO 100
    3 IF(X-XT(NTAB))6,5,4
    4 M=1
    5 N=NTAB
      GO TO 100
    6 I=N
      IF(X-XT(N))7,100,9
    7 I=I-1
      IF(X-XT(I))7,12,11
    8 I=I+1
    9 IF(X-XT(I+1))11,10,8
   10 N=I+1
      GO TO 100
   11 DELY=(YT(I+1)-YT(I))/(XT(I+1)-XT(I))*(X-XT(I))
   12 N=I
  100 TBLP=YT(N)+DELY
      RETURN
      END
$IBFTC DIVERG  DECK
      SUBROUTINE DIVERG
C        DIVERG CALCULATES D/DI,P,T,THETA AT DX INCREMENTS
C        ALONG STREAMLINE
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A(   10),DETAS ) , (A(   11),DX     ) , (A(   12),DXP    ),
     2(A(   13),ETAF  ) , (A(   30),URATIO) , (A(   33),XF     ),
     3(A(   34),XI    ) , (A(   35),XSIF  ) , (A(   37),XSI    ),
     4(A(   87),ETA   ) , (A(  137),ETAMAX) , (A(  187),TH8MX  ),
     5(A(  987),DODI  ) , (A( 1737),RORI  ) , (A( 2737),PRESSR),
     6(A( 5237),X     ) , (A( 5987),XSIX  ) , (A( 6737),ETAX   ),
     7(A( 7737),XSTAR ) , (A( 8487),PRESRX) , (A( 9237),THETAX),
     7(IA(    1),M    ),(IA(    2),N    ),(IA(    4),II   ),
     7(IA(    8),NPR  ),
```

```
     8(A(11587),P      ) , (A( 2737),THETAD)
      EQUIVALENCE (THETAX,THETAS)
C
      DIMENSION XSID(750,2),ETAD(750,2),XD(750,2),THETAD(750,2),XSIX(750
     1),ETAX(750),X(750),THETAX(750),ETA(50) ,XSI(50) ,
     2PRESRX(750),              PRESSR(50,50),XSTAR(750)
      DIMENSION DODI(750),RORI(750),THETAS(750),P(5)
      DIMENSION ETAMAX(50)
C
      REWIND 3
      WRITE(3)((PRESSR(I,J),I=1,M),J=1,N)
      NX=0
C        SAVE ORIGINAL STREAMLINE DATA
      I2=II
      DO 10 I=1,I2
      XSID(I,2)=XSIX(I)
      ETAD(I,2)=ETAX(I)
      XD(I,2)=X(I)
      THETAD(I,2)=THETAX(I)
      XSIS2=XF
   10 CONTINUE
C        CALCULATE STREAMLINE 1
      ETAZ=ETAF
      NXSV=1
      ETAMX=TBLP(XSI,ETAMAX,XSIF,M,NXSV)
      ETAF=ETAZ-DETAS
      IF(ETAF .GT. ETA(1)) GO TO 15
      ETAF=ETAZ+DETAS
      NX=1
   15 CONTINUE
      CALL STREAM
      I1=II
      DO 20 I=1,I1
      XSID(I,1)=XSIX(I)
      ETAD(I,1)=ETAX(I)
      XD(I,1)=X(I)
      THETAD(I,1)=THETAX(I)
      XSIS1=XF
   20 CONTINUE
      IF(XSIS1     .GT. XSIS2    ) GO TO 22
      GO TO 45
```

```
   22 CONTINUE                                                          24190457
      I1=I1+1                                                           24190458
      I=I1                                                              24190459
   30 CONTINUE                                                          24190460
      XSID(I,1)=XSID(I-1,1)                                             24190461
      ETAD(I,1)=ETAD(I-1,1)                                             24190462
      IF(I .LE. 2) GO TO 35                                             24190463
      I=I-1                                                             24190464
      GO TO 30                                                          24190465
   35 CONTINUE                                                          24190466
      XSID(1,1)=XSID(1,2)                                               24190467
      ETAD(1,1)=ETAD(2,1)                                               24190468
   45 CONTINUE                                                          24190469
      WRITE(6,6100)                                                     24190470
 6100 FORMAT(1H0,37HSTREAMLINE COORDINATES AND FLOW ANGLE //19H PRIMARY 24190471
     1STREAMLINE ,36X,20HAUXILIARY STREAMLINE//10X,3HXSI,9X,3HETA,9X,1HX24190472
     2,11X,5HTHETA,27X ,3HXSI,9X,3HETA,9X,1HX,11X,5HTHETA/)             24190473
      DO 12 I=1,I2                                                      24190474
      WRITE(6,6200)I,XSID(I,2),ETAD(I,2),XD(I,2),THETAD(I,2,,I,XSID(I,1)24190475
     1,ETAD(I,1),XD(I,1),THETAD(I,1)                                    24190476
 6200 FORMAT(1H ,I5,2X,4E12.5,10X,I5,2X,4E12.5)                         24190477
   12 CONTINUE                                                          24190478
C        CALCULATE P,T,THETA,D/DI AT DX INCREMENTS                      24190479
      NXS=1                                                             24190480
      NYS=1                                                             24190481
      NS2=1                                                             24190482
      NS1=1                                                             24190483
      XSTAR(1)=XI                                                       24190484
      XF=XD(I2,2)-3.*DX                                                 24190485
   44 CONTINUE                                                          24190486
      DO 51 I=1,I2                                                      24190487
      THETAS(I)=TBLP(XD(1,2),THETAD(1,2),XSTAR(I),I2,NS2)               24190488
   51 XSTAR(I+1)=XSTAR(I)+DX                                            24190489
      NS2=1                                                             24190490
      NI=0                                                              24190491
      DO 50 I=1,749                                                     24190492
      XSIS=TBLP(XD(1,2),XSID(1,2),XSTAR(I),I2,NS2)                      24190493
      ETAS= TBLP(XD(1,2),ETAD(1,2),XSTAR(I),I2,NS2)                     24190494
      ETA1=TBLP(XSID(1,1),ETAD(1,1),XSIS,I1,NS1)                        24190495
      ETA2=TBLP(XSID(1,2),ETAD(1,2),XSIS,I2,NS2)                        24190496
      IF(NX .GT. 0) GO TO 41                                            24190497
      IF(ETA1 .LT. ETA2) GO TO 42                                       24190498
```

```
   99 CONTINUE
      WRITE(6,6099)XSIS
 6099 FORMAT(1H0,26HSTREAMLINES CROSS AT XSI =  ,E12.5)
      XSTAR(1)=XSTAR(I)+DX
      XI=XSTAR(1)
      GO TO 44
   41 CONTINUE
      IF(ETA1 .GT. ETA2) GO TO 42
      GO TO 99
   42 CONTINUE
      IF(NI .GT. 0) GO TO 43
      REWIND 3
      READ(3)((PRESSR(K,J),K=1,M),J=1,N)
      REWIND 3
      NI=1
      COST1=COS(THETAS(I)/57.2958)
      ETA12=ETA2
      ETA11=ETA1
   43 CONTINUE
      RORI(I)=1.0
      DODI(I)=COS(THETAS(I)/57.2958)/COST1
      IF(NX .GT. 0) GO TO 46
      DODI(I)=DODI(I)*(ETA2-ETA1)/(ETA12-ETA11)
      GO TO 47
   46 CONTINUE
      DODI(I)=DODI(I)*(ETA1-ETA2)/(ETA11-ETA12)
   47 CONTINUE
      PRESRX(I)=DBTP(ETA,XSI,PRESSR,ETAS,XSIS,N,M,NYS,NXS)
      IF(XSTAR(I) .GT. XF)GO TO 60
      XSTAR(I+1)=XSTAR(I)+DX
      II=I
   50 CONTINUE
   60 CONTINUE
C        RESTORE ORIGINAL STREAMLINE
      DO 100 I=1,I2
      XSIX(I)=XSID(I,2)
      ETAX(I)=ETAD(I,2)
      X(I)=XD(I,2)
  100 CONTINUE
      IF(NPR .EQ. 6) GO TO 1890
      IF(URATIO .GT. 0.)GO TO 1890
```

```
      P(1)=PRESRX(1)
      P(2)=PRESRX(2)
      P(3)=PRESRX(3)
      DXP=DX
 1890 CONTINUE
      RETURN
      END
$IBFTC FIND     DECK
      SUBROUTINE FIND(ZA,Z,NZ,NZS)
      DIMENSION ZA(NZ)
      ML=0
C
      IF(Z-ZA(1))10,20,30
   10 ML=-1
   20 NZS=1
      GO TO 170
   30 IF(Z-ZA(NZ))60,50,40
   40 ML=1
   50 NZS=NZ
      GO TO 170
   60 IF(Z-ZA(NZS))80,170,120
   80 N=NZS-1
   90 IF(Z-ZA(N))100,160,160
  100 N=N-1
      GO TO 90
  120 N=NZS
  130 IF(Z-ZA(N))150,160,140
  140 N=N+1
      GO TO 130
  150 NZS=N-1
      GO TO 170
  160 NZS=N
  170 CONTINUE
      IF(ML)175,180,175
  175 CONTINUE
      WRITE(6,6010)Z,ZA(1),NZ,ZA(NZ)
 6010 FORMAT(1H0,32HVARIABLE NOT IN RANGE OF TABLE. /5H Z = ,E12.5,9H ZA
     1(1) = ,E12.5,4H ZA(,I3,4H) = ,E12.5)
  180 CONTINUE
      RETURN
      END
$IBFTC STREAM  DECK
```

```
      SUBROUTINE STREAM
C        CALCULATES STREAMLINE COORDINATES, PRESSURE AND DIVERGENCE
C        PARAMETERS
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A(   11),DX     ) , (A(   13),ETAF   ) , (A(   33),XF     ),
     2(A(   34),XI     ) , (A(   35),XSIF   ) , (A(   37),XSI    ),
     3(A(   87),ETA    ) , (A(  137),ETAMAX) , (A(  187),TH8MX  ),
     4(A(  237),THETAE) , (A( 5237),X      ) , (A( 5987),XSIX   ),
     5(A( 6737),ETAX   ) , (A( 9237),THETAS) , (A(    8),DELXSI),
     6(A( 8487),ETAS   ) , (A(10737),XSIS   ) , (A( 9987),XS     ),
     7(A( 7737),THETS ) ,
     6(IA(    1),M      ) , (IA(    2),N      ) , (IA(    4),II     )
      EQUIVALENCE (THETAX,THETAS)
      DIMENSION XSI(50),ETA(50),ETAMAX(50),TH8MX(50),THETAE(50,50)
      DIMENSION ETAS(750),XSIS(750),XS(750),ETAX(750),XSIX(750),X(750),
     1THETAS(750),THETS(750),THETAX(750)
      DIMENSION D(18)
      DATA (D(I),I=1,18)/18*6H******/
      RADIAN=57.2958
C
      IF(IT .EQ. 1) GO TO 90
      WRITE(6,6000) (XSI(I),I=1,9),D
 6000 FORMAT(19HOSTREAMLINE ANGLES //16H THETAE(XSI,ETA)//19X,3HXSI/16X,
     1 9E12.5/16X,18A6)
      DO 10 J=1,N
      WRITE (6,6010) ETA(J),(THETAE(I,J),I=1,9)
 6010 FORMAT(4H ETA,E11.4,2H *,9E12.5)
   10 CONTINUE
      M1=1
      M2=9
   15 CONTINUE
      IF(M .LE. M2) GO TO 80
      M1=M1+9
      M2=M2+9
      WRITE(6,6020) (XSI(I),I=M1,M2),D
 6020 FORMAT(19X,3HXSI/16X,9E12.5/16X,18A6)
      DO 20 J=1,N
      WRITE(6,6010) ETA(J),(THETAE(I,J),I=M1,M2)
   20 CONTINUE
      GO TO 15
```

```
   80 CONTINUE
      IT=1
      GO TO 100
   90 CONTINUE
      IT=0
C
  100 CONTINUE
      IX=0
      NX=1
      NY=1
      XSV=XSI(1)
      NXSV=1
      XSAV=XSI(1)
      YSAV=ETA(1)
      NXS=1
      NYS=1
      I=1
      XS1=0.
      XSIS1=XSIF
      ETAS1=ETAF
C
      CALL FIND(XSI,XSIS1,M,NX)
      CALL FIND(ETA,ETAS1,N,NY)
      THET1=THETAE(NX,NY+1)
      THET2=THETAE(NX+1,NY+1)
      THET3=THETAE(NX+1,NY)
      THET4=THETAE(NX,NY)
      TH8S1=DBTP(ETA,XSI,THETAE,ETAS1,XSIS1,N,M,NYS,NXS)
      THETS(1)=TH8S1
      XSIS(1)=XSIF
      ETAS(1)=ETAF
      XS(1)=0.
  105 CONTINUE
      DXSI=DX*COS(TH8S1/RADIAN)
      DETA=DX*SIN(TH8S1/RADIAN)
      IF(TH8S1 .LE. 0.) DXSI=DX
      IF(TH8S1 .LE. 0.) DETA=0.
      ETAS2=ETAS1-DETA
      IF(TH8S1 .LE. 0.) ETAS2=ETAS1
      XSIS2=XSIS1-DXSI
      XS2=XS1+DX
      ETAD=XSIS2
```

```
      IF(ETAS2 .GE. ETA(1)) GO TO 130
      ETAS2=ETA(1)
      XSIS2=XSIS1-(ETAS1-ETA(1))/TAN(TH8S1/RADIAN)
      XS2=XS1+(ETAS1-ETA(1))/SIN(TH8S1/RADIAN)
      IX=1
      GO TO 135
  130 CONTINUE
      IX=0
  135 CONTINUE
      IF(XSIS2 .GE. XSI(1)) GO TO 145
      XSIS2=XSI(1)
      ETAS2=ETAS1-(XSIS1-XSI(1))*TAN(TH8S1/RADIAN)
      XS2=XS1+(XSIS1-XSI(1))/COS(TH8S1/RADIAN)
      IX=1
  145 CONTINUE
      IF(IX.GT.0)GO TO 2000
C
      CALL FIND(XSI,XSIS2,M,NX)
      XSI1=XSI(NX)
      XSI2=XSI(NX+1)
      XSI3=XSI(NX+1)
      XSI4=XSI(NX)
      DC=XSI2-XSI1
      CALL FIND(ETA,ETAS2,N,NY)
      ETA1=ETA(NY+1)
      ETA2=ETA(NY+1)
      ETA3=ETA(NY)
      ETA4=ETA(NY)
      DE=ETA1-ETA4
C
      THET1=THETAE(NX,NY+1)
      THET2=THETAE(NX+1,NY+1)
      THET3=THETAE(NX+1,NY)
      THET4=THETAE(NX,NY)
      ETAMX1=TBLP(XSI,ETAMAX,XSI1,M,NXSV)
      ETAMX2=TBLP(XSI,ETAMAX,XSI2,M,NXSV)
      ETAMXS=TBLP(XSI,ETAMAX,XSIS2,M,NXSV)
      SLOPE=ETAMX2-ETAMX1
      IF(ETAS2-ETAMXS)150,150,1900
  150 CONTINUE
      IF(SLOPE)152,152,200
```

```
  152 CONTINUE
      IF(ETA2-ETAMX2)1100,1100,154
  154 CONTINUE
      IF(ETA1-ETAMX1)156,156,158
  156 CONTINUE
      IF(ETA3-ETAMX2)1600,1600,1700
  158 CONTINUE
      IF(ETA4-ETAMX1)160,160,999
  160 CONTINUE
      IF(ETA3-ETAMX2)1400,1400,1800
C
  200 CONTINUE
      IF(ETA1-ETAMX1)1100,1100,210
  210 CONTINUE
      IF(ETA2-ETAMX2)220,220,230
  220 CONTINUE
      IF(ETA4-ETAMX1)1200,1200,1300
  230 CONTINUE
      IF(ETA3-ETAMX2)240,240,999
  240 CONTINUE
      IF(ETA4-ETAMX1)1400,1400,1500
C
C         ROUTINES TO CALCULATE THETAE AT XSIS2, ETAS2
C
 1100 CONTINUE
C         ALL POINTS ON BODY
      TH8S2=DBTP(ETA,XSI,THETAE,ETAS2,XSIS2,N,M,NY,NXS)
      GO TO 2000
C
 1200 CONTINUE
C         POINTS 2-3-4 ON BODY - POSITIVE SLOPE
      TH83=THET4+(XSIS2-XSI4)/DC*(THET3-THET4)
      IF(ETAMXS .LE. ETA1) GO TO 1210
      THM1=TBLP(ETAMAX,TH8MX,ETA2,M,NXSV)
      XSIM=TBLP(ETAMAX,XSI,ETA2,M,NXSV)
      TH81=THM1+(XSIS2-XSIM)/(XSI2-XSIM)*(THET2-THM1)
      TH8S2=TH83+(ETAS2-ETA3)/DE*(TH81-TH83)
      GO TO 1220
 1210 CONTINUE
      TH81=TBLP(XSI,TH8MX,XSIS2,M,NXSV)
      ETAM=TBLP(XSI,ETAMAX,XSIS2,M,NXSV)
      TH8S2=TH83+(ETAS2-ETA3)/(ETAM-ETA3)*(TH81-TH83)
```

```
 1220 CONTINUE                                                  24190748
      GO TO 2000                                                24190749
C                                                               24190750
 1300 CONTINUE                                                  24190751
C        POINTS 2-3 ON BODY - POSITIVE SLOPE                    24190752
      THM=TBLP(ETAMAX,TH8MX,ETA3,M,NXSV)                        24190753
      XSIM=TBLP(ETAMAX,XSI,ETA1,M,NXSV)                         24190754
      TH83=THM+(XSIS2-XSIM)/(XSI3-XSIM)*(THET3-THM)             24190755
      IF(ETAMXS .LE. ETA1) GO TO 1310                           24190756
      THM=TBLP(ETAMAX,TH8MX,ETA1,M,NXSV)                        24190757
      XSIM=TBLP(ETAMAX,XSI,ETA1,M,NXSV)                         24190758
      TH81=THM+(XSIS2-XSIM)/(XSI2-XSIM)*(THET2-THM)             24190759
      TH8S2=TH83+(ETAS2-ETA3)/DE*(TH81-TH83)                    24190760
      GO TO 1320                                                24190761
 1310 CONTINUE                                                  24190762
      TH81=TBLP(XSI,TH8MX,XSIS2,M,NXSV)                         24190763
      ETAM=TBLP(XSI,ETAMAX,XSIS2,M,NXSV)                        24190764
      TH8S2=TH83+(ETAS2-ETA3)/(ETAM-ETA3)*(TH81-TH83)           24190765
 1320 CONTINUE                                                  24190766
      GO TO 2000                                                24190767
C                                                               24190768
 1400 CONTINUE                                                  24190769
C        POINTS 3-4 ON BODY - POSITIVE SLOPE                    24190770
      TH83=THET4+(XSIS2-XSI4)/DC*(THET3-THET4)                  24190771
      TH81=TBLP(XSI,TH8MX,XSIS2,M,NXSV)                         24190772
      ETAM=TBLP(XSI,ETAMAX,XSIS2,M,NXSV)                        24190773
      TH8S2=TH83+(ETAS2-ETA3)/(ETAM-ETA3)*(TH81-TH83)           24190774
      GO TO 2000                                                24190775
 1500 CONTINUE                                                  24190776
C        ONE POINT ON BODY (POINT 3)                            24190777
C        POSITIVE SLOPE                                         24190778
      TH81=TBLP(ETAMAX,TH8MX,ETAS2,M,NXSV)                      24190779
      TH82=TBLP(XSI,TH8MX,XSIS2,M,NXSV)                         24190780
      ALPHA=ATAN((ETAS2-ETA3)/(XSI3-XSIS2))                     24190781
      ETAMX=TBLP(XSI,ETAMAX,XSIS2,M,NXSV)                       24190782
      XSIMX=TBLP(ETAMAX,XSI,ETAS2,M,NXSV)                       24190783
      CD=XSIS2-XSIMX                                            24190784
      BD=ETAMX-ETAS2                                            24190785
      GAM=ATAN(BD/CD)                                           24190786
      DEL=3.14159-ALPHA-GAM                                     24190787
      CE=SIN(ALPHA)*CD/SIN(DEL)                                 24190788
```

```
      BC=SQRT(CD**2+BD**2)
      TH83=TH81+CE/BC*(TH82-TH81)
      ED=SIN(GAM)/SIN(DEL)*CD
      AD=(XSI3-XSIS2)/COS(ALPHA)
      EA=ED+AD
      TH84=THETAE(NX+1,NY)
      TH8S2=TH83+ED/EA*(TH84-TH83)
      GO TO 2000
C
 1600 CONTINUE
      TH81=THET4+(XSIS2-XSI4)/DC*(THET3-THET4)
      TH82=TBLP(XSI,TH8MX,XSIS2,M,NXSV)
      TH8S2=TH81+(ETAS2-ETA3)/(ETAMXS-ETA3)*(TH82-TH81)
      GO TO 2000
C
 1700 CONTINUE
C        NEGATIVE SLOPE
C        TWO POINTS ON BODY (POINTS 1 AND 4)
      TH81=THET4+(ETAS2-ETA4)/DE*(THET1-THET4)
      TH82=TBLP(ETAMAX,TH8MX,ETAMXS,M,NXSV)
      XSIMX=TBLP(ETAMAX,XSI,ETAS2,M,NXSV)
      TH8S2=TH81+(XSIS2-XSI4)/(XSIMX-XSI4)*(TH82-TH81)
      GO TO 2000
C
 1800 CONTINUE
C        ONE POINT ON BODY (POINT 4)
      TH81=TBLP(ETAMAX,TH8MX,ETAS2,M,NXSV)
      TH82=TBLP(XSI,TH8MX,XSIS2,M,NXSV)
      ALPHA=ATAN((ETAS2-ETA4)/(XSIS2-XSI4))
      ETAMX=TBLP(XSI,ETAMAX,XSIS2,M,NXSV)
      XSIMX=TBLP(ETAMAX,XSI,ETAS2,M,NXSV)
      CD=XSIMX-XSIS2
      BD=ETAMX-ETAS2
      GAM=ATAN(BD/CD)
      DEL=3.14159-ALPHA-GAM
      BC=SQRT(BD**2+CD**2)
      CE=SIN(ALPHA)/SIN(DEL)*CD
      TH83=TH81+CE/BC*(TH82-TH81)
      ED=SIN(GAM)/SIN(DEL)*CD
      AD=(XSIS2-XSI4)/COS(ALPHA)
      EA=ED+AD
      TH84=THETAE(NX,NY)
```

```
      TH8S2=TH83+ED/EA*(TH84-TH83)
      GO TO 2000
C
 1900 CONTINUE
C         * POINT IS OFF OF BODY
 1710 CONTINUE
      XSIMX=TBLP(ETAMAX,XSI,ETAS2,M,NXSV)
      ETAMX=ETAS2+(XSIMX-XSIS2)*TAN(TH8S1/RADIAN)
      IF(ABS(ETAMX-ETAS2)-1.E-06)1750,1750,1720
 1720 CONTINUE
      XSIS2=XSIMX
      ETAS2=ETAMX
      GO TO 1710
 1750 CONTINUE
      XSIS2=XSIMX
      ETAS2=ETAMX
      TH8S2=TBLP(ETAMAX,TH8MX,ETAMX,M,NXSV)
      I=I+1
      XS(I)=XS1+(ETAS1-ETAS2)/SIN(TH8S1/RADIAN)
      XSIS(I)=XSIS2
      ETAS(I)=ETAS2
      THETS(I)=TH8S1
      IF(ETAD .LE. ETA(1)+.00001)XSIS(I)=0.
      IF(ETAD .LE. ETA(1)+.00001)ETAS(I)=0.
      GO TO 2100
C
 2000 CONTINUE
      I=I+1
      XSIS(I)=XSIS2
      XS(I)=XS2
      ETAS(I)=ETAS2
      THETS(I)=TH8S2
      IF(IX .GT. 0) THETS(I)=TH8S1
      IF(IX .GT. 0) GO TO 2100
C
      XSIS1=XSIS2
      XS1=XS2
      ETAS1=ETAS2
      XSIS1=XSIS2
      TH8S1=TH8S2
      GO TO 105
```

```
C
 2100 CONTINUE
C         REORDER STREAMLINE COORDINATES
      XSIX(1)=XSIS(I)
      ETAX(1)=ETAS(I)
      X(1)=0.
      THETAX(1)=THETS(I)
      J=1
      DO 2200 J=2,I
      K=I-J+1
      XSIX(J)=XSIS(K)
      ETAX(J)=ETAS(K)
      X(J)=XS(I)-XS(K)
      THETAX(J)=THETS(K)
 2200 CONTINUE
 3000 CONTINUE
      XF=XSIS(I)
      II=I
C
      GO TO 5000
  999 CONTINUE
      WRITE(6,6999)ETAMX1,ETAMX2,ETA3,ETA4
 6999 FORMAT(1H0,31HERROR IN STREAMLINE CALCULATION/11H ETAMAX1 = ,E12.5
     1,11H ETAMAX2 = ,E12.5,8H ETA3 = ,E12.5,8H ETA4 = ,E12.5)
      STOP
 5000 CONTINUE
      RETURN
      END
$IBFTC WALLT1  DECK
      SUBROUTINE WALLT1(XS,TWS)
C        CALCULATES TW(X)
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A( 7737),XSTAR ) , (A(10737),TW     ) , (IA(    4),II     ),
     2(IA(    9),ITC    )
      DIMENSION XS(50),TWS(50),TW(750),XSTAR(750)
      NSV=1
      DO 10 I=1,II
      TW(I)=TBLP(XS,TWS,XSTAR(I),ITC,NSV)
   10 CONTINUE
      RETURN
      END
```

```
$IBFTCTWALLT2  DECK
      SUBROUTINE WALLT2(MT,NT,XSIT,ETAT)
C        CALCULATES TW(XSI,ETA)
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A( 5237),TWALL ) , (A( 6737),ETAX   ) , (A( 5987),XSIX   ),
     2(A(10737),TW    ) , (IA(    4),II    ) , (IA(    9),ITC    )
      DIMENSION XSIT(50),ETAT(50),TW(750)
      DIMENSION ETAX(750),XSIX(750)
      NXS=1
      NYS=1
      IF(ITC .LT. 2) GO TO 20
      DO 10 I=1,II
      TW(I)=DBTP(ETAT,XSIT,TWALL,ETAX(I),XSIX(I),NT,MT,NYS,NXS)
   10 CONTINUE
      GO TO 30
   20 CONTINUE
      WRITE(6,6010)
 6010 FORMAT(23H0ERROR IN WALLT2. ITC =I2)
   30 CONTINUE
      RETURN
      END
$IBFTC BLKDTA  DECK
      BLOCK DATA
      COMMON/BLKDAT/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
C
      DATA R,PSL,ACCG,RADIAN,EPS/
     11716.,2116.2,32.1739,57.29578,.000001/
      DATA(PRESST(I),I .,7)/
     1-4.,-3.,-2.,-1.,0.,1.,2./
      DATA((ZTT(I,J),I=1,7),J=1,26)/
     17*111.  , 7*223.  , 7*333.  , 7*444.  , 7*555.  , 7*833.  ,
     27*1110. , 7*1387. , 7*1665. , 7*1944. , 7*2222. , 7*2500. ,
     37*2778. , 7*3056. , 7*3333. , 7*3889. , 7*4444. , 7*5000. ,
     47*5556. , 7*6667. , 7*7778. , 7*8889. , 7*10000., 7*11111.,
     57*13889., 7*16667./
      DATA ((AYRTOT(I,J),I=1,7),J=1,9)/
     17*1.42,7*2.84,7*4.26,7*5.68,7*7.10,7*11.0,7*14.8,7*19.1,7*23.7/
      DATA ((AYRTOT(I,J),I=1,7),J=10,18)/
     1 30.04,  28.73,  28.02,  28.02,  28.02,  28.02,  28.02,
```

```
2 40.10,  36.57,  34.26,  33.05,  32.84,  32.83,  32.83,                24190955
3 53.61,  46.59,  42.07,  40.05,  38.55,  37.54,  37.54,                24190956
4 69.12,  59.63,  52.12,  47.60,  45.08,  43.27,  43.25,                24190957
5 80.09,  72.11,  63.13,  56.60,  52.09,  49.57,  47.48,                24190958
6 88.06,  82.58,  75.10,  66.11,  60.09,  56.37,  54.47,                24190959
7 99.12,  96.60,  93.09,  86.11,  77.12,  70.12,  67.60,                24190960
8116.16, 111.13, 105.09, 101.09,  93.60,  86.12,  81.10,                24190961
9138.27, 129.19, 118.17, 114.13, 108.12, 102.12,  94.61/                24190962
  DATA ((AYRTOT(I,J),I=1,7),J=19,26)/                                   24190963
1168.43, 150.34, 137.25, 128.19, 121.17, 115.13, 107.14,                24190964
2248.47, 212.44, 184.40, 164.31, 152.24, 139.20, 132.16,                24190965
3323.51, 282.49, 247.45, 213.41, 190.31, 170.27, 157.22,                24190966
4394.45, 350.49, 310.45, 270.45, 233.40, 207.31, 187.27,                24190967
5450.27, 410.45, 366.49, 325.48, 282.43, 245.38, 220.32,                24190968
6480.26, 460.28, 420.36, 378.44, 330.44, 287.42, 255.36,                24190969
7545.63, 530.40, 510.33, 490.33, 440.45, 393.49, 347.45,                24190970
8670.77, 610.49, 575.40, 555.40, 530.55, 490.59, 437.55/                24190971
  DATA((OMEGAT(I,J),I=1,7),J=1,10)/                                     24190972
17*14.47, 7*13.68, 7*12.46, 7*11.54, 7*10.80, 7*9.34, 7*8.316,          24190973
27*7.560, 7*6.980, 7*6.534/                                             24190974
  DATA((OMEGAT(I,J),I=1,7),J=11,20)/                                    24190975
16.030, 6.102, 6.138, 6.138, 6.138, 6.138, 6.138, 5.580, 5.670,         24190976
25.760, 5.832, 5.832, 5.832, 5.832, 5.202, 5.310, 5.400, 5.472,         24190977
35.544, 5.544, 5.544, 4.896, 4.986, 5.094, 5.184, 5.256, 5.310,         24190978
45.310, 4.644, 4.734, 4.824, 4.914, 5.004, 5.058, 5.058, 4.248,         24190979
54.320, 4.392, 4.482, 4.590, 4.680, 4.680, 3.960, 4.014, 4.086,         24190980
64.158, 4.230, 4.338, 4.410, 3.726, 3.780, 3.834, 3.906, 3.960,         24190981
74.068, 4.140, 3.546, 3.600, 3.654, 3.708, 3.780, 3.834, 3.906,         24190982
83.186, 3.276, 3.330, 3.384, 3.438, 3.510, 3.582/                       24190983
  DATA((OMEGAT(I,J),I=1,7),J=21,26)/                                    24190984
12.934, 3.024, 3.078, 3.132, 3.204, 3.276, 3.330, 2.736, 2.808,         24190985
22.880, 2.952, 3.006, 3.060, 3.132, 2.574, 2.664, 2.718, 2.790,         24190986
32.844, 2.916, 2.970, 2.448, 2.520, 2.592, 2.664, 2.718, 2.790,         24190987
42.844, 2.214, 2.286, 2.358, 2.430, 2.484, 2.556, 2.592, 2.070,         24190988
52.142, 2.214, 2.268, 2.304, 2.376, 2.430/                              24190989
C                                                                       24190990
      END                                                               24190991
$IBFTC DERV    DECK                                                     24190992
      SUBROUTINE DERV(V,W,X,Y,Z,DBYDX,DX,ITYPE)                         24190993
      GO TO (10,20,30,40,50),ITYPE                                      24190994
   10 DBYDX=(-V+W)/DX                                                   24190995
      GO TC 60                                                          24190996
```

```
   20 DBYDX=(-V+X)/(2.*DX)
      GO TO 60
   30 DBYDX=(-W+Y)/(2.*DX)
      GO TO 60
   40 DBYDX=(-X+Z)/(2.*DX)
      GO TO 60
   50 DBYDX=(-Y+Z)/DX
   60 RETURN
      END
$IBFTC EBAR    DECK
      SUBROUTINE EBAR(AYEW,AYESL,H,SIGR,EXPK,POPSL,OMEGE,OMEGS,GAMC,GAMO
     1,EBARL,EBART)
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      CALL SOMEGA(AYESL,POPSL,ZTESL,OMESL)
      AYEAV=.5*(AYEW+AYESL)
      PR=PSL*POPSL
      ROMUT=PR*OMEGS/R
      ROMUE=PR*OMEGE/R
      AYEMC=AYEAV+.2*(H-AYESL)*SIGR**SQRT(ROMUT/ROMUE)
      CALL SOMEGA(AYEMC,POPSL,ZTMC,OMMC)
      SIGC=ZTMC/ZTESL
C
      CALL SOMEGA(AYEAV,POPSL,ZTMO,OMMO)
      SIGO=ZTMO/ZTESL
C
      FSC=(SIGC-.294)*SIGR**.355/.4018
      FSO=(SIGO-.294)*SIGR**.355/.4018
      FKC=(2.*SIGC)**EXPK
      FKO=(2.*SIGO)**EXPK
      GAMC=.71764*FKC*(SQRT(1.+FSC)-1.)
      GAMO=.71764*FKO*(SQRT(1.+FSO)-1.)
C
      EBARL=1.+GAMC
      EBART=(1.+.76519*GAMO)*((1.+GAMC)/(1.+GAMO))**4
      RETURN
      END
$IBFTC IWALL   DECK
      SUBROUTINE IWALL(TW,POPSL,AYEW)
 1105 CONTINUE
      PPLOG=ALOG10(POPSL)
      TWK=TW/1.8
```

```
      IF(TWK-300.)1110,1110,1115
 1110 CONTINUE
      AYEW=.432*TWK
      GO TO 1130
 1115 CONTINUE
      IF(TWK-1800.)1120,1120,1125
 1120 CONTINUE
      AYEW=.432*TWK+.0000382*(TWK-300.)**2
      GO TO 1130
 1125 CONTINUE
      CALL TABLE6(TWK,PPLOG,AYEW)
 1130 CONTINUE
      AYEW=25031.3*AYEW
      RETURN
      END
$IBFTC JAYELL  DECK
      SUBROUTINE JAYELL(AYEW,H,AYEE,ZTE,ZTS,SIGR,BETAS,POPSL,XJL)
C
      AYEMS=.5*(AYEW+H)
      CALL SOMEGA(AYEMS,POPSL,ZTMS,OMMS)
      SIGS=ZTMS/ZTS
      AYEWHL=AYEE/H+SQRT(SIGR)*(1.-AYEE/H)
      FSS=(SIGS-.294)*SIGR**.355*AYEWHL/.4018
      CPES=(ZTE/ZTS)/(AYEE/H)
      IF(BETAS)10,20,20
   10 CONTINUE
      FPM=-1.
      J=-1
      GO TO 30
   20 CONTINUE
      FPM=1.
      J=1
   30 CONTINUE
      FBETAS=(1.+2.*CPES)*ABS(BETAS)/(2.*CPES+BETAS*FPM)
      XJL=(1.+.71764*SQRT(1.+FBETAS*FSS)-.71764)**J
      RETURN
      END
$IBFTC MUZERO  DECK
      SUBROUTINE MUZERO(OMEGR,ZTR,AYER,AYEE,H,PSTPSL,XMUO,XL)
      XMUO=OMEGR*ZTR*(H/AYER)**1.5*(ZTR+200.)/((H/AYER)*ZTR+200.)
      AYEEC=AYEE/25031.6/33.86
      PPLOG=ALOG10(PSTPSL)
```

```
      CALL TABLE7(AYEEC,PPLOG,AYEDIE)
      IF(AYEDIE)10,20,20
   10 CONTINUE
      AYEDIE=0.
   20 CONTINUE
      XL=1.+.19*AYEDIE*AYEE/H
      RETURN
      END
$IBFTC RORMUR  DECK
      SUBROUTINE RORMUR(P,OMEGS,OMEGE,OMEGW,OMEGR,ROMUR,ZTR,AYER,SIGR)
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      PPLOG=ALOG10(P)
      OMSE=OMEGS/OMEGE
      XKS=OMSE*(1.005/(.005+OMSE**7))**(1./14.)
      XKEXP=.8125+.1875*EXP(-XKS*OMEGE/OMEGW)
      XK=XKS**XKEXP
      F1=XK*OMEGE/OMEGW
      OMRW=F1**.875*(1.2/(.2+F1**5))**.1
      PR=P*PSL
      OMEGR=OMRW*OMEGW
      ROMUR=OMEGR/R*PR
      OMRK=1.8*OMEGR
      IF(OMRK-3.585E-10)30,10,10
   10 CONTINUE
      IF(OMRK-8.03E-10)20,20,30
   20 CONTINUE
      ZTRK=23213.13E-20/OMRK**2*(1.+SQRT(1.-.478656E18*OMRK**2))**2
      GO TO 40
   30 CONTINUE
      CALL TABL19(OMRK,PPLOG,ZTRK)
   40 CONTINUE
      CALL TABL20(ZTRK,PPLOG,AYERTO)
      AYER=847559.8*AYERTO
      ZTR=ZTRK*1.8
      CALL TABL14(ZTR,SIGR)
      RETURN
      END
$IBFTC SOMEGA  DECK
C     ROUTINE TO OBTAIN OMEGA AND ZT
C     ENTHALPY IN FT**2/SEC**2 AND PRESSURE IN ATMOSPHERES
      SUBROUTINE SOMEGA(ENTH,PRESS,ZT,OMEGA)
```

```
      ENTHC=ENTH/25031.3
      PPLOG=.434294*ALOG(PRESS)
   20 IF(ENTHC-180.)21,21,22
   21 ZT=ENTHC/.432
      GO TO 40
   22 ENTHC=ENTHC/33.86
      CALL TABLE2(ENTHC,PPLOG,ZT,CHECK)
      IF(CHECK-1.)30,40,30
   30 CONTINUE
      WRITE(6,6000) PPLOG,ENTHC,ZT
 6000 FORMAT(1H0,23HERROR IN TABLE2 LOOK-UP,5X,3E12.5)
   40 IF(ZT-111.11)41,41,42
   41 OMEGA=14.454E-10
      GO TO 60
   42 IF(ZT-2000.)43,43,44
   43 OMEGA=SQRT(1.8*ZT/200.)*(400./(1.8*ZT+200.))*14.454E-10
      GO TO 60
   44 CALL TABL18(ZT,PPLOG,OMEGA,CHECK)
      IF(CHECK-1.)50,60,50
   50 WRITE(6,6100) PPLOG,ZT,OMEGA
 6100 FORMAT(1H0,23HERROR IN TABL18 LOOK-UP,5X,3E12.5)
   60 ZT=ZT*1.8
      OMEGA=OMEGA/1.8
   70 RETURN
      END
$IBFTC TABLE2  DECK
C     ZT=F(I/RT,P)
      SUBROUTINE TABLE2(AYES,PEDGEL,TS,CHECK)
      COMMON/BLKDAT/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      DIMENSION L(13)
      DATA KS2/1/
      IF(KS2-1)1,1,10
    1 CONTINUE
      L(1)=LOC(L(1))
      L(2)=LOC(PRESST(1))
      L(3)=1
      L(4)=7
      L(5)=2
      L(6)=2
      L(7)=XTAB(0)
      L(9)=LOC(AYRTOT(1,1))
```

```
      L(10)=LOC(ZTT(1,1))
      L(11)=7
      L(12)=7
      L(13)=26
      KS2=2
   10 L(8)=0
      TS=DTAB(AYES,PEDGEL,L(1))
      CHECK=L(8)
      RETURN
      END
$IBFTC TABLE6  DECK
C     TABLE6 I=F(T,P)
      SUBROUTINE TABLE6(TW,PEDGEL,AYEW)
      DIMENSION TWT(19),AYE(8,19),PRESS(8),L(13)
      DATA KS6/1/
      DATA(TWT(J),J=1,19)/
     10.,500.,1000.,1500.,1800.,2000.,2200.,2400.,2600.,2800.,3000.,
     23200.,3400.,3600.,3800.,4000.,4200.,4300.,4400./
      DATA(PRESS(I),I=1,8)/
     1-5.,-4.,-3.,-2.,-1.,0.,1.,2./
      DATA((AYE(I,J),J=1,19),I=1,4)/
     10., 240., 450., 700., 870., 1320., 2067., 2668., 2893., 3044.,
     23180., 3374., 3680., 4253., 5376., 7350.,10000.,11600.,12800.,
     10., 240., 450., 700., 870., 1097., 1510., 2183., 2730., 2982.,
     23140., 3285., 3468., 3735., 4176., 4940., 6205., 7150., 8150.,
     10., 240., 450., 700., 870., 1022., 1242., 1630., 2202., 2728.,
     23044., 3227., 3385., 3555., 3778., 4105., 4607., 4990., 5399.,
     10., 240., 450., 700., 870., 1003., 1150., 1360., 1688., 2156.,
     22655., 3030., 3276., 3456., 3627., 3823., 4068., 4220., 4405./
      DATA((AYE(I,J),J=1,19),I=5,8)/
     10., 240., 450., 700., 870.,  994., 1120., 1267., 1464., 1727.,
     22080., 2496., 2895., 3220., 3467., 3659., 3847., 3990., 4051.,
     10., 240., 450., 700., 870.,  992., 1108., 1237., 1381., 1554.,
     21764., 2026., 2339., 2683., 3032., 3325., 3582., 3680., 3805.,
     10., 240., 450., 700., 870.,  990., 1105., 1225., 1355., 1494.,
     21647., 1822., 2023., 2250., 2512., 2783., 3063., 3230., 3343.,
     10., 240., 450., 700., 870.,  988., 1102., 1221., 1346., 1475.,
     21607., 1751., 1903., 2068., 2245., 2436., 2637., 2750., 2857./
      IF(KS6-1)1,1,10
    1 CONTINUE
      L(1)=LOC(L(1))
```

```
      L(2)=LOC(PRESS(1))                                                24191204
      L(3)=1                                                            24191205
      L(4)=8                                                            24191206
      L(5)=1                                                            24191207
      L(6)=1                                                            24191208
      L(7)=XTAB(0)                                                      24191209
      L(9)=LOC(TWT(1))                                                  24191210
      L(10)=LOC(AYE(1,1))                                               24191211
      L(11)=1                                                           24191212
      L(12)=8                                                           24191213
      L(13)=19                                                          24191214
      KS6=1                                                             24191215
   10 L(8)=0                                                            24191216
      AYEW=DTAB(TW,PEDGEL,L(1))                                         24191217
      RETURN                                                            24191218
      END                                                               24191219
$IBFTC TABLE7  DECK                                                     24191220
C     TABLE7 ID/IE=F(I/RT,P)                                            24191221
      SUBROUTINE TABLE7(AYEE,PEDGEL,AYEAYE)                             24191222
      DIMENSION AYDAYE(7,26),L(13)                                      24191223
      COMMON/BLKDAT/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)       24191224
      DATA KS7/1/                                                       24191225
      DATA ((AYDAYE(I,J),I=1,7),J=1,16)/                                24191226
     17*0., 7*0., 7*0., 7*0., 7*0., 7*0., 7*0., 7*.008,                 24191227
     2.020, .020, .018, .017, .024, .025, .026,                         24191228
     3.045, .032, .028, .033, .036, .036, .037,                         24191229
     4.239, .134, .091, .058, .049, .048, .047,                         24191230
     5.327, .316, .236, .109, .058, .053, .053,                         24191231
     6.367, .376, .314, .163, .084, .067, .063,                         24191232
     7.412, .425, .366, .233, .128, .087, .071,                         24191233
     8.459, .459, .396, .294, .178, .113, .083,                         24191234
     9.466, .485, .435, .367, .283, .182, .126/                         24191235
      DATA ((AYDAYE(I,J),I=1,7),J=17,25)/                               24191236
     1.490, .475, .452, .397, .338, .255, .173,                         24191237
     2.537, .477, .447, .409, .365, .301, .216,                         24191238
     3.607, .517, .452, .411, .375, .323, .251,                         24191239
     4.724, .647, .546, .460, .400, .354, .281,                         24191240
     5.757, .719, .660, .555, .462, .392, .314,                         24191241
     6.761, .744, .708, .635, .526, .438, .347,                         24191242
     7.760, .754, .728, .673, .588, .484, .387,                         24191243
     8.757, .750, .736, .690, .624, .532, .430,                         24191244
     9.745, .731, .721, .697, .661, .596, .523/                         24191245
```

```
      DATA(AYDAYE(I,26),I=1,7)/                                         24191246
     1.748, .723, .704, .686, .664, .618, .559/                         24191247
      IF(KS7-1)1,1,10                                                   24191248
    1 CONTINUE                                                          24191249
      L(1)=LOC(L(1))                                                    24191250
      L(2)=LOC(PRESST(1))                                               24191251
      L(3)=1                                                            24191252
      L(4)=7                                                            24191253
      L(5)=2                                                            24191254
      L(6)=2                                                            24191255
      L(7)=XTAB(0)                                                      24191256
      L(9)=LOC(AYRTOT(1,1))                                             24191257
      L(10)=LOC(AYDAYE(1,1))                                            24191258
      L(11)=7                                                           24191259
      L(12)=7                                                           24191260
      L(13)=26                                                          24191261
      KS7=2                                                             24191262
   10 L(8)=0                                                            24191263
      AYEAYE    AB(AYEE,PEDGEL,L(1))                                    24191264
      RETURN                                                            24191265
      END                                                               24191266
$IBFTC TABL14  DECK                                                     24191267
C     TABL14 SIG=F(ZT)                                                  24191268
      SUBROUTINE TABL14(ZTW,SIGT)                                       24191269
      DIMENSION SIGW(35),ZTT(35),L(8)                                   24191270
      DATA KS14/1/                                                      24191271
      DATA(ZTT(I),I=1,35)/                                              24191272
     1    0.,   200.,   400.,   600.,   800.,  1000.,  1200.,  1400.,   24191273
     2 1600.,  1800.,  2000.,  2500.,  3000.,  3500.,  4000.,  4200.,   24191274
     3 4600.,  5000.,  6000.,  7000.,  8000.,  9000., 10000., 11000.,   24191275
     412000., 13000., 14000., 15000., 16000., 17000., 18000., 19000.,   24191276
     520000., 21000., 22000./                                           24191277
      DATA(SIGW(J),J=1,35)/                                             24191278
     1 .770, .770, .728, .699, .684, .680, .682, .687, .695, .706, .716,24191279
     2 .740, .758, .770, .776, .777, .774, .770, .761, .750, .742, .733,24191280
     3 .724, .715, .708, .701, .695, .690, .687, .685, .684, .685, .689,24191281
     4 .694, .704 /                                                     24191282
      IF(KS14-1)1,1,10                                                  24191283
    1 CONTINUE                                                          24191284
      L(1)=LOC(L(1))                                                    24191285
      L(2)=LOC(ZTT(1))                                                  24191286
```

```
      L(3)=LOC(SIGW(1))
      L(4)=1
      L(5)=1
      L(6)=2
      L(7)=35
      KS14=2
   10 L(8)=0
      SIGT=TAB(ZTW,L(1))
      RETURN
      END
$IBFTC TABL18  DECK
C     TABL18 OMEGA=F(ZT,P)
      SUBROUTINE TABL18(ZT,PRESSL,OMEGA,CHECK)
      COMMON/BLKDAT/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      DIMENSION L(13)
      DATA KS18/1/
      IF(KS18-1)1,1,10
    1 CONTINUE
      L(1)=LOC(L(1))
      L(2)=LOC(PRESST(1))
      L(3)=1
      L(4)=7
      L(5)=1
      L(6)=1
      L(7)=XTAB(0)
      L(9)=LOC(ZTT(1,1))
      L(10)=LOC(OMEGAT(1,1))
      L(11)=7
      L(12)=7
      L(13)=26
      KS18=2
   10 L(8)=0
      OMEGA=(DTAB(ZT,PRESSL,L(1)))*1.E-10
      RETURN
      END
$IBFTC TABL19  DECK
C     TABL19 ZT=F(OMEGA,P)
      SUBROUTINE TABL19(OMEGA,PRESSL,ZT)
      COMMON/BLKDAT/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      DIMENSION L(13)
```

```
      DATA KS19/1/
      IF(KS19-1)1,1,10
    1 CONTINUE
      L(1)=LOC(L(1))
      L(2)=LOC(PRESST(1))
      L(3)=1
      L(4)=7
      L(5)=1
      L(6)=1
      L(7)=XTAB(0)
      L(9)=LOC(OMEGAT(1,1))
      L(10)=LOC(ZTT(1,1))
      L(11)=7
      L(12)=7
      L(13)=26
      KS19=2
   10 L(8)=0
      OMEGA = OMEGA/1.E-10
      ZT=DTAB(OMEGA,PRESSL,L(1))
      RETURN
      END
$IBFTCTTABL20  DECK
C     TABL20 I/RT=F(ZT,P)
      SUBROUTINE TABL20(ZT,PRESSL,AYERTO)
      COMMON/BLKDAT/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      DIMENSION L(13)
      DATA KS20/1/
      IF(KS20-1)1,1,10
    1 CONTINUE
      L(1)=LOC(L(1))
      L(2)=LOC(PRESST(1))
      L(3)=1
      L(4)=7
      L(5)=1
      L(6)=1
      L(7)=XTAB(0)
      L(9)=LOC(ZTT(1,1))
      L(10)=LOC(AYRTOT(1,1))
      L(11)=7
      L(12)=7
```

```
      L(13)=26                                                          24191370
      KS20=2                                                            24191371
   10 L(8)=0                                                            24191372
      AYERTO=DTAB(ZT,PRESSL,L(1))                                       24191373
      RETURN                                                            24191374
      END                                                               24191375
$IBFTC DELTA    DECK                                                    24191376
      SUBROUTINE DELTA                                                  24191377
C        CALCULATES INPUTS TO STREAM FOR DELTA WING                     24191378
C        RESTRICTIONS                                                   24191379
C          60 GE SWEEP LE 80                                            24191380
C          6 GE MACH LE 22                                              24191381
C          ANGLE OF ATTACK GT 0.                                        24191382
      COMMON/Q/A(11600),IA(10)                                          24191383
      EQUIVALENCE                                                       24191384
     1(A(    1),ACH    ) , (A(    2),ALPHA ) , (A(    7),DELETA),       24191385
     2(A(    8),DELXSI) , (A(   13),ETAF   ) , (A(   14),ETAI   ),      24191386
     3(A(   15),H      ) , (A(   18),PINF   ) , (A(   20),POPSL  ),     24191387
     4(A(   22),RADIUS) , (A(   25),RHO    ) , (A(   27),SWEEP  ),      24191388
     5(A(   30),URATIO) , (A(   31),VEL    ) , (A(   35),XSIF   ),      24191389
     6(A(   36),XSII   ) , (A(   37),XSI    ) , (A(   87),ETA    ),     24191390
     7(A(  137),ETAMAX) , (A(  187),TH8MX  ) , (A(  237),THETAE),       24191391
     8(A( 2737),PRESSR) , (IA(    1),M     ) , (IA(    2),N      )      24191392
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS                               24191393
      DIMENSION XSI(50),ETAMAX(50),TH8MX(50),ETA(50),THETAE(50,50),     24191394
     1PRESSR(50,50),PHIT1(25),XN1(25),SWEEP1(50),XNT2(50,50),PHIT2(50)  24191395
      DATA NT1,(PHIT1(I),XN1(I),I=1,21)/  21,                           24191396
     1.10, .095,  .15, .122,  .20, .143,  .25, .183,  .30, .222,        24191397
     2.35, .258,  .40, .292,  .45, .327,  .50, .388,  .55, .450,        24191398
     3.60, .473,  .65, .480,  .70, .479,  .75, .475,  .80, .467,        24191399
     4.85, .457,  .90, .449,  .95, .439,  1.0, .430,  1.1, .413,        24191400
     51.2, .398 /                                                       24191401
      DATA NTI,NTJ,(PHIT2(J),J=1,50)/ 5,50,                             24191402
     11.2 , 1.3 , 1.4 , 1.5 , 1.6 , 1.7 , 1.8 , 1.9 , 2.0 , 2.1 , 2.2 , 24191403
     22.3 , 2.4 , 2.5 , 2.6 , 2.7 , 2.8 , 2.9 , 3.0 , 3.1 , 3.2 , 3.3 , 24191404
     33.4 , 3.5 , 3.6 , 3.7 , 3.8 , 3.9 , 4.0 , 4.1 , 4.2 , 4.3 , 4.4 , 24191405
     44.5 , 4.6 , 4.7 , 4.8 , 4.9 , 5.0 , 5.1 , 5.2 , 5.3 , 5.4 , 5.5 , 24191406
     55.6 , 5.8 , 6.0 , 6.2 , 6.4 , 6.6/                                24191407
      DATA SWEEPT(1),(XNT2(1,J),J=1,50)/ 60.,                           24191408
     1.398, .390, .395, .421, .480, .550, .645, .760, 1.00, 41*1.00 /   24191409
      DATA SWEEPT(2),(XNT2(2,J),J=1,50)/ 65.,                           24191410
     1.398, .387, .377, .380, .400, .426, .460, .498, .542, .597, .663, 24191411
```

```
     2.755, .900, 1.00, 36*1.00 /
      DATA SWEEPT(3),(XNT2(3,J),J=1,50)/ 70.,
     1.398, .387, .377, .370, .372, .390, .398, .421, .445, .470, .498,
     2.525, .577, .587, .620, .664, .715, .780, .860, 1.00, 30*1.00 /
      DATA SWEEPT(4),(XNT2(4,J),J=1,50)/ 75.,
     1.398, .387, .377, .370, .366, .372, .378, .385, .394, .405, .418,
     2.432, .443, .460, .476, .495, .510, .528, .545, .565, .585, .610,
     3.635, .660, .690, .720, .755, .790, .835, .880, .930, 19*1.00 /
      DATA SWEEPT(5),(XNT2(5,J),J=1,50)/ 80.,
     1.398, .387, .377, .370, .365, .362, .364, .366, .372, .375, .380,
     2.384, .392, .398, .403, .412, .421, .430, .438, .443, .453, .461,
     3.470, .479, .487, .497, .505, .517, .524, .536, .546, .560, .570,
     4.583, .598, .612, .627, .642, .660, .677, .694, .710, .730, .750,
     5.768, .805, .850, .898, .945, 1.0/
C          SHARP DELTA WING
C
      DEL=ALPHA/57.2958
      CPSD=0.
      IF(DEL .GT. 0.) CPSD=1.05+SQRT(1.1025+4./ACH**2/SIN(DEL)**2)-
     11.278*SIN(DEL)**2/ACH**0.6
      P2=PINF*(0.7*ACH**2*CPSD*SIN(DEL)**2+1.)
      U=1.-.5863*DEL**2
      URATIO=U
      P=P2/PSL
      AIE=H-(U*VEL)**2/2.
      CALL SOMEGA(AIE,P,ZTE,OMEGAE)
      RHO2=P2/1716./ZTE
      RR2=RHO/RHO2
      ZETA=SQRT((DEL+1.22*RR2*(TAN(DEL)/TAN(SWEEP/RADIAN))
     1**.566)**2+ASIN(1./ACH)**2)
      ACHN2=(ACH*SIN(ZETA))**2
      PHI=(1./(90.-SWEEP))*ATAN(SQRT(((1.+5./ACH**2*(ACHN2-1.)/ACHN2)/
     1U**2-1.)/6.))*RADIAN
      GAM=PHI**4/(PHI**4+.2)
      THET=(90.-SWEEP)*(1.+PHI)*GAM
      IF(PHI-1.2)10,10,20
   10 CONTINUE
      NN=1
      XNCL=TBLP(PHIT1,XN1,PHI,NT1,NN)
      GO TO 30
   20 CONTINUE
```

```
      NX=1                                                              24191453
      NY=1                                                              24191454
      XNCL =DBTP(PHIT2,SWEEPT,XNT2,PHI,SWEEP,NTJ,NTI,NX,NY)             24191455
   30 CONTINUE                                                          24191456
      XNCL=(1.+PHI)*XNCL/((ACHN2+1.)/ACHN2)                             24191457
      DELN=(90.-SWEEP)*XNCL                                             24191458
      THDN=THET-DELN                                                    24191459
      TAND=TAN((90.-SWEEP)/RADIAN)                                      24191460
      I=1                                                               24191461
      XSI(I)=XSII                                                       24191462
   40 CONTINUE                                                          24191463
      J=1                                                               24191464
      ETAMAX(I)=XSI(I)*TAND                                             24191465
      TH8MX(I)=DELN+THDN                                                24191466
      ETA(J)=ETAI                                                       24191467
   50 CONTINUE                                                          24191468
      AOAM=TAN(ETA(J)/XSI(I))/TAND                                      24191469
      THETAE(I,J)=DELN*AOAM+THDN*AOAM**9                                24191470
      PRESSR(I,J)=P2/PSL/POPSL                                          24191471
      J=J+1                                                             24191472
      ETA(J)=ETA(J-1)+DELETA                                            24191473
      IF(ETA(J) .LE. ETAMAX(I)+DELETA) GO TO 50                         24191474
      I=I+1                                                             24191475
      XSI(I)=XSI(I-1)+DELXSI                                            24191476
   60 CONTINUE                                                          24191477
      IF(XSI(I) .GT. XSIF+DELXSI) GO TO 70                              24191478
      GO TO 40                                                          24191479
   70 CONTINUE                                                          24191480
      M=I-1                                                             24191481
      N=J-1                                                             24191482
  100 CONTINUE                                                          24191483
      RETURN                                                            24191484
      END                                                               24191485
$IBFTC CYLIND  DECK                                                     24191486
      SUBROUTINE CYLIND                                                 24191487
C        CALCULATES INPUTS TO STREAM FOR SWEPT CYLINDER                 24191488
      COMMON/Q/A(11600),IA(10)                                          24191489
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS                               24191490
      EQUIVALENCE                                                       24191491
     1(A(    1),ACH    ) , (A(   11),DX     ) , (A(   12),DXP    ),     24191492
     2(A(   18),PINF   ) , (A(   20),POPSL  ) , (A(   22),RADIUS),      24191493
     3(A(   27),SWEEP  ) , (A(   30),URATIO) , (A(   31),VEL    ),      24191494
```

```
     4(A(   37),XSI   ) , (A(   87),ETA    ) , (A(  137),ETAMAX),
     5(A(  187),TH8MX ) , (A(  237),THETAE) , (A( 5237),TWALL ),
     6(A( 8487),PRESRX) , (A( 9987),UE     ) , (A(10737),TW    ),
     7(A(11587),P     ) , (A( 2737),PRESSR) , (UE      ,V     )
     8,(IA(  1),M     ) , (IA(   2),N      ) , (IA(   4),II    )
C
      DIMENSION XSI(50),ETA(50),ETAMAX(50),TH8MX(50),THETAE(50,50),
     1PRESSR(50,50),UE(750),V(750),PRESRX(750),TW(750),TWALL(50,50)
      DIMENSION TH8T(50),PPT2T(50)
      DIMENSION P(5)
      DIMENSION XSTAR(750),TWX(750)
C
C
      DATA NT,(TH8T(I),PPT2T(I),I=1,44)/ 44,
     1 0.,1.00000,  .2, .99998782,  .4, .99995126,  .6, .99989034,
     1  .8, .99980,  1.0, .99970, 1.2, .99956,  1.4, .99940,
     1 1.6, .99922,  1.8, .99902, 2.0, .99878,  3.0, .99726,
     2  4., .99512,   5., .99239,  6., .98907,   7., .98516,
     3  8., .98063,   9., .97553, 10., .96985,  11., .963597,
     4 12., .95677,  13., .949397,14., .94148,  15., .93302,
     5 16., .92402,  17., .91451, 18., .90452,  19., .89401,
     1                    20.,.887, 24.,.840, 28.,.795, 32.,.732, 36.,
     2.670, 40.,.608, 44.,.546, 48.,.485, 52.,.426, 56.,.370, 60.,.317,
     365.,.258, 70.,.207, 80.,.124, 90.,.0775, 100.,.05/
C
      NXS=1
      NXSV=1
      XSV=DX
      USIN=VEL*SIN(SWEEP/RADIAN)
      ACHN=ACH*COS(SWEEP/RADIAN)
      DETA=.01*RADIUS
      PT2=(1.2*ACHN**2)**3.5*(6./(7.*ACHN**2-1.))**2.5*PINF
      XSTAR(1)=0.
      DO 20 I=1,750
      THETR=RADIAN*XSTAR(I)/RADIUS
      IF(THETR-100.)10,30,30
   10 CONTINUE
      POPT2=TBLP(TH8T,PPT2T,THETR,NT,NXS)
      PRESRX(I)=POPT2*PT2/(POPSL*PSL)
      TWX(I)=TBLP(ETA,TW,XSTAR(I),N,NXSV)
      XSTAR(I+1)=XSTAR(I)+DETA
```

```
      II=I                                                              24191536
   20 CONTINUE                                                          24191537
      WRITE(6,6100)THETR,ETA(II)                                        24191538
 6100 FORMAT(1H0,67HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SU24191539
     1BROUTINE CYLIND /15H THETAR-ETA(II) ,5X,2E12.5//26H RUN TERMINATED24191540
     2 BY PROGRAM )                                                     24191541
      STOP                                                              24191542
   30 CONTINUE                                                          24191543
      II=II+1                                                           24191544
      XSTAR(II)=XSTAR(II-1)+DETA                                        24191545
      PRESRX(II)=PRESRX(II-1)                                           24191546
      TWX(II)=TWX(II-1)                                                 24191547
      DX=DETA                                                           24191548
      URATIO=0.                                                         24191549
      P(1)=1.0                                                          24191550
      P(2)=.99998782                                                    24191551
      P(3)=.99995126                                                    24191552
      DXP=.00349                                                        24191553
      CALL FLOW                                                         24191554
      M=2                                                               24191555
      N=II/4                                                            24191556
      J=1                                                               24191557
      DO 35 I=1,N                                                       24191558
      PRESRX(I)=PRESRX(J)                                               24191559
      TWX(I)=TWX(J)                                                     24191560
      V(I)=V(J)                                                         24191561
      ETA(I)=XSTAR(J)                                                   24191562
      J=J+4                                                             24191563
   35 CONTINUE                                                          24191564
      DO 40 I=1,9                                                       24191565
      XSI(I)=0.                                                         24191566
      DO 40 J=1,N                                                       24191567
      THETAE(I,J)=0.                                                    24191568
   40 CONTINUE                                                          24191569
      DO 60 I=1,M                                                       24191570
      DO 50 J=1,N                                                       24191571
      THETAE(I,J)=ATAN(V(J)/USIN)*RADIAN                                24191572
      PRESSR(I,J)=PRESRX(J)                                             24191573
   50 CONTINUE                                                          24191574
   60 CONTINUE                                                          24191575
      XSI(1)=0.                                                         24191576
      XSI(2)=1.E+6                                                      24191577
```

```
      ETAMAX(1)=ETA(N)
      ETAMAX(2)=ETA(N)
      TH8MX(1)=THETAE(1,N)
      TH8MX(2)=THETAE(1,N)
      DX=XSV
      URATIO=    SIN(SWEEP/57.2958)
      RETURN
      END
$IBFTC FLOW    DECK
      SUBROUTINE FLOW
C        FLOW CALCULATES THE EXTERNAL FLOW AND STAGNATION
C     PROPERTIES
C
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A(    11),DX     ) , (A(    12),DXP    ) , (A(    15),H      ),
     2(A(    20),POPSL  ) , (A(    30),URATIO) , (A(    31),VEL    ),
     3(A(    27),SWEEP  ) , (A(   237),POPSLX) , (A(  2737),AIEE   ),
     4(A(  3487),OMEGAE) , (A(  4237),ZTEX   ) , (A(  7737),XSTAR  ),
     5(A(  8487),PRESRX) , (A(  9987),UE     ) , (A(11587),P      ),
     5(IA(     4),II     ) , (IA(     8),NPR    )
      DIMENSION PRESRX(750),POPSLX(750),UE(750),AIEE(750),
     1ZTEX(750),OMEGAE(750),XSTAR(750)
      DIMENSION P(5)
C
C
C
C
      DO 780 J=1,II
      POPSLX(J)=POPSL*PRESRX(J)
  780 CONTINUE
C        CALCULATE VELOCITY, EDGE ENTHALPY, AND ZT ARRAYS
  800 CONTINUE
      KK=2
      UEINF=.5*URATIO**2
      UE(1)=URATIO*VEL
      U=UE(1)
      IF(NPR .EQ. 6) U=VEL*SIN(SWEEP/57.2958)
      AIEE(1)= H-.5*U**2
      CALL SOMEGA(AIEE(1),POPSLX(1),ZTEX(1),OMEGAE(1))
```

```
      ZTE=ZTEX(1)
      IF(URATIO)805,805,810
  805 CONTINUE
      SECDRV=(P(1)-2.*P(2)+P(3))/DXP**2
      UEOUI=DX/VEL*SQRT(-R*ZTE*SECDRV)
      UEINF=.5*UEOUI**2
      UE(2)=VEL*UEOUI
      U=UE(2)
      AIEE(2)=H-.5*U**2
      IF(NPR .EQ. 6)AIEE(2)=AIEE(2)-.5*(VEL*SIN(SWEEP/57.2958))**2
      CALL SOMEGA(AIEE(2),POPSLX(2),ZTEX(2),OMEGAE(2))
      ZTE=ZTEX(2)
      KK=KK+1
  810 CONTINUE
      DO 865 I=KK,II
      IF(I-3)815,820,825
  815 CONTINUE
      CALL DERV(PRESRX(I-1),PRESRX(I),PRESRX(I+1),PRESRX(I+2),
     1PRESRX(I+3),DBYDX1,DX,1)
      CALL DERV(PRESRX(I-1),PRESRX(I),PRESRX(I+1),PRESRX(I+2),
     1PRESRX(I+3),DBYDX2,DX,2)
      GO TO 845
  820 CONTINUE
      CALL DERV(PRESRX(I-2),PRESRX(I-1),PRESRX(I),PRESRX(I+1),
     1PRESRX(I+2),DBYDX1,DX,2)
      CALL DERV(PRESRX(I-2),PRESRX(I-1),0.,PRESRX(I+1),PRESRX(I+2),
     1DBYDX2,DX,3)
      GO TO 845
  825 CONTINUE
      IF(I-II+1)830,835,840
  830 CONTINUE
      CALL DERV(PRESRX(I-3),PRESRX(I-2),0.,PRESRX(I),PRESRX(I+1),
     1DBYDX1,DX,3)
      CALL DERV(PRESRX(I-2),PRESRX(I-1),0.,PRESRX(I+1),PRESRX(I+2),
     1DBYDX2,DX,3)
      GO TO 845
  835 CONTINUE
      CALL DERV(PRESRX(I-3),PRESRX(I-2),0.,PRESRX(I),PRESRX(I+1),
     1DBYDX1,DX,3)
      CALL DERV(PRESRX(I-3),PRESRX(I-2),PRESRX(I-1),PRESRX(I),
     1PRESRX(I+1),DBYDX2,DX,4)
      GO TO 845
```

```
 840  CONTINUE
      CALL DERV(PRESRX(I-4),PRESRX(I-3),PRESRX(I-2),PRESRX(I-1),
     1PRESRX(I),DBYDX1,DX,4)
      CALL DERV(PRESRX(I-4),PRESRX(I-3),PRESRX(I-2),PRESRX(I-1),
     1PRESRX(I),DBYDX2,DX,5)
 845  CONTINUE
      DBYDX=(DBYDX1+DBYDX2)/2.
      JJ = 1
      DO 855 NO=1,10
      UEINF2=UEINF1
      UEINF1 = UEINF-R*ZTE*DBYDX*DX/(VEL**2*((PRESRX(I-1)+PRESRX(I))/2.)
     1)
      AIEE(I)=H-UEINF1*VEL**2
      IF(NPR .EQ. 6)AIEE(I)=AIEE(I)-.5*(VEL*SIN(SWEEP/57.2958))**2
      CALL SOMEGA(AIEE(I),POPSLX(I),ZTE,OMEGA)
      IF(JJ)847,850,847
  847 CONTINUE
      JJ=0
      GO TO 855
 850  CONTINUE
      IF(ABS(UEINF1-UEINF2)/UEINF1-EPS)860,855,855
 855  CONTINUE
      WRITE(6,6005)UEINF1,UEINF2
 6005 FORMAT(1H0,62HCONVERGENCE NOT ESTABLISHED IN VELOCITY CALCULATION.
     1 UEINF1 = ,E12.5,10H UEINF2 = ,E12.5)
 860  CONTINUE
      UEINF=UEINF1
      UE(I)=VEL*SQRT(2.*UEINF)
      ZTEX(I)=ZTE
      OMEGAE(I)=OMEGA
 865  CONTINUE
      IF(NPR .EQ. 6) NPR=100
C
      RETURN
      END
$IBFTC AXI2D   DECK
      SUBROUTINE AXI2D
C        CALCULATES INPUTS TO QTRAN FOR TWO-DIMENSIONAL AND
C        AXISYMMETRIC BODIES
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      COMMON/Q/A(11600),IA(10)
```

```
      EQUIVALENCE
     1(A(    1),ACH    ) , (A(    5),BEQXL ) , (A(    6),BEQXT ),
     2(A(   11),DX     ) , (A(   15),H     ) , (A(   16),HO    ),
     3(A(   18),PINF   ) , (A(   20),POPSL ) , (A(   30),URATIO),
     4(A(   31),VEL    ) , (A(   34),XI    ) , (A(   37),XSI   ),
     5(A(   87),Y      ) , (A(  987),DODI  ) , (A( 1737),RORI  ),
     6(A( 2737),AIEE   ) , (A( 3487),OMEGAE) , (A( 4237),ZTEX  ),
     7(A( 5237),X      ) , (A( 5987),XSIX  ) , (A( 6737),ETAX  ),
     8(A( 7737),XSTAR  ) , (A( 9237),THETAX) , (A( 9987),UE    ),
     9(A(  237),POPSLX) , (IA(    1),M     ) , (IA(    4),II    ),
     1(IA(    8),NPR    )
      DIMENSION XSI(50),Y(50),XSTAR(750),POPSLX(750),UE(750),AIEE(750),
     1ZTEX(750),OMEGAE(750),DODI(750),RORI(750)
      DIMENSION YS(750),PRESS(750),XS(50)
      DIMENSION DELS(50),XSIS(50)
      DIMENSION XST(50),PREST(50)
      DIMENSION X(750),XSIX(750),ETAX(750),THETAX(750)
C
      NXSV=1
      XSIS(1)=XSI(1)
      YS(1)=Y(1)
      XS(1)=XI
      XSTAR(1)=XI
      IF(NPR-2)120,120,110
  110 CONTINUE
      DODI(1)=1.0
      RORI(1)=1.0
      GO TO 130
  120 CONTINUE
      DODI(1)=Y(1)
      RORI(1)=Y(1)
  130 CONTINUE
      DO 100 I=2,M
      XSIS(I)=0.5*(XSI(I)+XSI(I-1))
      YS(I)=0.5*(Y(I)+Y(I-1))
      XS(I)=XS(I-1)+SQRT((YS(I)-YS(I-1))**2+(XSIS(I)-XSIS(I-1))**2)
      DELS(I)=ATAN((Y(I)-Y(I-1))/(XSI(I)-XSI(I-1)))
      IF(XSI(I) .EQ. XSI(I-1)) DELS(I)=1.5707963
      DEL=DELS(I)
      IF(DELS(I))10,10,20
   10 CONTINUE
      CPSD=0.
```

```
      GO TO 30
   20 CONTINUE
      CPSD=1.05+SQRT(1.1025+4./ACH**2/SIN(DEL)**2)-1.278*SIN(DEL)**2/
     1ACH**0.6
   30 CONTINUE
      P2=PINF*(0.7*ACH**2*CPSD*SIN(DEL)**2+1.)
      PRESS(I)=(P2/PSL)
  100 CONTINUE
      PRESS(1)=PRESS(3)-(PRESS(3)-PRESS(2))*(XSIS(3)-XSIS(1))/(XSIS(3)-
     1XSIS(2))
      XSIS(M+1)=XSI(M)
      YS(M+1)=Y(M)
      XS(M+1)=XS(M)+SQRT((YS(M+1)-YS(M))**2+(XSIS(M+1)-XSIS(M))**2)
      PRESS(M+1)=PRESS(M)+(PRESS(M)-PRESS(M-1))*(XSIS(M+1)-XSIS(M))/
     1(XSIS(M)-XSIS(M-1))
      M=M+1
      IF(PRESS(M) .LT. PINF/PSL) PRESS(M)=PINF/PSL
      IF(PRESS(M) .GT. POPSL) PRESS(M)=POPSL
      IF(PRESS(1) .LT. PINF/PSL) PRESS(1)=PINF/PSL
      DO 200 I=1,749
      XSTAR(I+1)=XSTAR(I   )+DX
      X(I)=XSTAR(I)
      XSIX(I)=TBLP(XS,XSIS,XSTAR(I),M,NXSV)
      ETAX(I)=TBLP(XS,YS,XSTAR(I),M,NXSV)
      THETAX(I)=TBLP(XS,DELS,XSTAR(I),M,NXSV)
      THETAX(I)=57.2958*THETAX(I)
      IF(XSTAR(I+1)-XS(M))170,210,210
  170 CONTINUE
      POPSLX(I)=TBLP(XS,PRESS,XSTAR(I),M,NXSV)
      DEL=TBLP(XS,DELS,XSTAR(I),M,NXSV)
      IF(PRESS(1)-.528*POPSL)40,40,50
   40 CONTINUE
      UE(I)=VEL*SQRT(1.-(POPSLX(I)/PINF*PSL-1.)/1.4/ACH**2)
      GO TO 60
   50 CONTINUE
      IF(I .GT. 1) GO TO 58
      P=.528*POPSL
      DO 55 K=1,M
      J=M-K+1
      PREST(K)=PRESS(J)
      XST(K)=XS(J)
```

```
   55 CONTINUE
      XSONIC=RBLP(PREST,XST,P,M,1)
      SONIC=SQRT(H /3.)
   58 CONTINUE
      IF(XSTAR(I)-XSONIC)70,70,80
   70 CONTINUE
      UE(I)=SONIC/XSONIC*XSTAR(I)
      GO TO 90
   80 CONTINUE
      UE(I)=SONIC*SQRT(6.*(1*-(POPSLX(I)/POPSL)**.285714))
   90 CONTINUE
   60 CONTINUE
      AIEE(I)=H-UE(I)**2/2.
      CALL SOMEGA(AIEE(I),POPSLX(I),ZTEX(I),OMEGAE(I))
      IF(NPR-2)150,150,140
  140 CONTINUE
      DODI(I)=1.
      RORI(I)=1.
      GO TO 160
  150 CONTINUE
      DODI(I)=TBLP(XS,YS,XSTAR(I),M,NXSV)
      RORI(I)=DODI(I)
  160 CONTINUE
      II=I
  200 CONTINUE
      THETAX(I)=THETAX(2)
      WRITE(6,6100)XSTAR(I),M,XS(M),DX
 6100 FORMAT(1H0,66HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SU
     1BROUTINE AXI2D /20H XSTAR(I)-M-XS(M)-DX ,5X,E12.5,I3,2X,2E12.5)
      WRITE(6,6120)
 6120 FORMAT(1H0,25HRUN TERMINATED BY PROGRAM)
      STOP
  210 CONTINUE
      URATIO=UE(I)/VEL
C
      IF(NPR-2)224,224,221
  224 CONTINUE
      WRITE(6,6150)
 6150  FORMAT(1H0,19HAXISYMMETRIC OUTPUT)
      GO TO 223
  221 CONTINUE
      WRITE(6,6160)
```

```
 6160 FORMAT(1H0,22HTWO-DIMENSIONAL OUTPUT)
  223 CONTINUE
      WRITE(6,6200)XI,DX
 6200 FORMAT(1H0,10X,5HXI = ,E12.5,6H DX = ,E12.5//12X,3HXSI,9X,1HY/)
      M=M-1
      DO 222 I=1,M
      WRITE(6,6300)XSI(I),Y(I)
 6300 FORMAT(1H ,10X,12E11.3)
  222 CONTINUE
      M=M+1
      RETURN
      END
$IBFTC HEMI    DECK
      SUBROUTINE HEMI
C        CALCULATES INPUTS TO FLOW FOR HEMISPHERE
C
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A(    1),ACH    ) , (A(   11),DX     ) , (A(   12),DXP    ),
     2(A(   18),PINF   ) , (A(   20),POPSL  ) , (A(   22),RADIUS),
     3(A(   34),XI     ) , (A(  987),DODI   ) , (A( 1737),RORI   ),
     4(A( 5237),X      ) , (A( 5987),XSIX   ) , (A( 6737),ETAX   ),
     5(A( 7737),XSTAR  ) , (A( 9237),THETAX) , (A(11587),P      ),
     6(A(   30),URATIO) , (A( 8487),PRESRX) , (IA(    4),II     )
      DIMENSION XSTAR(750),PRESRX(750),DODI(750),RORI(750)
      DIMENSION TH8T(50),PPT2T(50)
      DIMENSION P(5)
      DIMENSIONX(750),XSIX(750),ETAX(750),THETAX(750)
C
      DATA NT,(TH8T(I),PPT2T(I),I=1,44)/ 44,
     1 0.,1.00000,  .2, .99998782,  .4, .99995126,  .6, .99989034,
     1  .8, .99980,  1.0, .99970, 1.2, .99956,  1.4, .99940,
     1 1.6, .99922,  1.8, .99902, 2.0, .99878,  3.0, .99726,
     2  4., .99512,   5., .99239,  6., .98907,   7., .98516,
     3  8., .98063,   9., .97553, 10., .96985,  11., .963597,
     4 12., .95677,  13., .949397,14., .94148,  15., .93302,
     5 16., .92402,  17., .91451, 18., .90452,  19., .89401,
     6 20., .88302,  24., .838,   28., .785,    32., .720,
     7 36., .653,    40., .585,   44., .517,    48., .450,
     8 52., .384,    56., .320,   60., .262,    65., .200,
```

```
     9 70.,.147,    80.,.076,   90.,.045,  100.,.4/
C
      NXSV=1
      XI=0.
      P(1)=1.0
      P(2)=.99998782
      P(3)=.99995126
      DXP=.00349
      DX=.00300*RADIUS
      XSTAR(1)=XI
      DO 100 I=1,750
      THET=RADIAN*XSTAR(I)/RADIUS
      X(I)=XSTAR(I)
      XSIX(I)=0.
      ETAX(I)=0.
      THETAX(I)=90.-THET
      IF(THETAX .LT. 0.) THETAX(I)=0.
      IF(THET-100.)50,50,110
   50 CONTINUE
      POPT2=TBLP(TH8T,PPT2T,THET,NT,NXSV)
      PRESRX(I)=POPT2
      RORI(I)=SIN(XSTAR(I)/RADIUS)
      DODI(I)=RORI(I)
      XSTAR(I+1)=XSTAR(I)+DX
      II=I
  100 CONTINUE
      WRITE(6,6100)THET,XSTAR(I),DX
 6100 FORMAT(1H0,65HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SU
     1BROUTINE HEMI /15H THETA-XSTAR-DX ,5X,3E12.5)
      WRITE(6,6120)
 6120 FORMAT(1H0,25HRUN TERMINATED BY PROGRAM)
      STOP
  110 CONTINUE
      URATIO=0.
C
      RETURN
      END
$IBFTC QTRAN   DECK
      SUBROUTINE QTRAN
C        QTRN CALCULATES HEAT TRANSFER DATA GIVEN THE
C     STREAMLINE FUNCTIONS
      COMMON/Q/A(11600),IA(10)
```

```
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      EQUIVALENCE
     1(A(    4),AYEWO ) , (A(    5),BEQXL ) , (A(    6),BEQXT ),
     2(A(   11),DX    ) , (A(   15),H     ) , (A(   16),HO    ),
     3(A(   17),HREFT ) , (A(   21),QINF  ) , (A(   26),RTRANS),
     4(A(   31),VEL   ) , (A(   32),VELP  ) , (A(   34),XI    ),
     5(A(  237),POPSLX) , (A(  987),DODI  ) , (A( 1737),RORI  ),
     6(A( 2737),AIEE  ) , (A( 3487),OMEGAE) , (A( 4237),ZTEX  ),
     7(A( 5237),X     ) , (A( 5987),XSIX  ) , (A( 6737),ETAX  ),
     8(A( 7737),XSTAR ) , (A( 9237),THETAS) , (A( 9987),UE    ),
     9(A(10737),TW    ) , (A(11487),XPO   ) , (A(   21),RADIUS),
     1(IA(   3),KF    ) , (IA(   4),II    ) , (IA(   8),NPR   ),
     2(A( 2730),AYEW  ) , (A(  232),SL    ) , (A(  980),ST    ),
     3(IA(   9),ITC   )
      DIMENSION AYEW(750),SL(750),ST(750)
      DIMENSION XSTAR(750),POPSLX(750),UE(750),AIEE(750),ZTEX(750),
     1OMEGAE(750),THETAS(750),DODI(750),RORI(750),TW(750),XPO(100)
      DIMENSION X(750),XSIX(750),ETAX(750)
      INTEGER GSAVE,AA
C
      FINTRP(Q0FL,Q1FL,Q2FL,Q3FL,Q4FL)=Q3FL+(Q4FL-Q3FL)/
     1(Q1FL-Q0FL)*(Q2FL-Q0FL)
C
      WRITE(6,6200)
 6200 FORMAT(1H1,3X,1HX,11X,5HP/PSL,7X,2HUE,10X,4HDODI,8X,4HRORI,8X,2HTW
     1,10X,3HZTE,9X,6HOMEGAE,6X,4HAIEE,8X,4HAYEW/
     24X,3HXSI,9X,5HBEQXL,7X,5HXEQXL,7X,6HAYEAWL,6X,2HJL,10X,5HEBARL,7X,
     32HHL,10X,5HQLISO,7X,5HQDOTL,7X,4HCFEL/
     44X,3HETA,9X,5HBEQXT,7X,5HXEQXT,7X,6HAYEAWT,6X,2HN  ,10X,5HEBART,7X,
     52HHT,10X,5HQTISO,7X,5HQDOTT,7X,4HCFET/
     64X,5HTHETA,7X,3HZTR,9X,3HMUO,9X,3HFXQ,9X,3HFXS,9X,3HRRQ,9X,3HRRS,
     79X,6HOMEGAR,6X,6HTHETAL,6X,6HTHETAT)
      DO 100 K=1,KF
      KK=K
      IF(XPO(K) .GT. (XSTAR(II)-3.*DX)) GO TO 110
  100 CONTINUE
      GO TO 120
  110 CONTINUE
      KF=KK-1
  120 CONTINUE
C        INITIALIZATION
```

```
 900  CONTINUE
      NXSV=1
      MXSV=1
      I=0
      K=1
      ITRAN=0
      ITR=0
      IPRINT=0
      INTERP=0
      GSAVE=0
      COUNT=0.
      PHIL=0.
      PHIT=0.
      BETAS=0.5
      CTR=0.
      IP=XI/DX
      IP=IP+II
      XTRANS=0.
C
C        POINT FOR REPEATED LOOPING AS X IS INCREMENTED BY DX
C
 1001 CONTINUE
      I=I+1
 1100 CONTINUE
      CALL IWALL(TW(I),POPSLX(I),AYEW(I))
      CALL SOMEGA(H,POPSLX(I),ZTS,OMEGAS)
      CALL SOMEGA(AYEW(I),POPSLX(I),ZTW,OMEGAW)
      CALL RORMUR(POPSLX(I),OMEGAS,OMEGAE(I),OMEGAW,OMEGAR,ROMUR,ZTR,
     1AYER,SIGR)
C
C        VELOCITY AND DELTA DERIVATIVE
 1200 CONTINUE
      IF(I-2)1210,1220,1230
 1210 CONTINUE
      CALL DERV(DODI(I),DODI(I+1),DODI(I+2),DODI(I+3),DODI(I+4),DDELDX,D
     1X,1)
      CALL DERV(UE(I),UE(I+1),UE(I+2),UE(I+3),UE(I+4),DUEDX,DX,1)
      GO TO 1270
 1220 CONTINUE
      CALL DERV(DODI(I-1),DODI(I),DODI(I+1),DODI(I+2),DODI(I+3),DDELDX,D
     1X,2)
      CALL DERV(UE(I-1),UE(I),UE(I+1),UE(I+2),UE(I+3),DUEDX,DX,2)
```

```
      GO TO 1270
 1230 CONTINUE
      IF(I-II+1)1240,1250,1260
 1240 CONTINUE
      CALL DERV(DODI(I-2),DODI(I-1),0.,DODI(I+1),DODI(I+2),DDELDX,DX,3)
      CALL DERV(UE(I-2),UE(I-1),0.,UE(I+1),UE(I+2),DUEDX,DX,3)
      GO TO 1270
 1250 CONTINUE
      CALL DERV(DODI(I-3),DODI(I-2),DODI(I-1),DODI(I),DODI(I+1),DDELDX,D
     1X,4)
      CALL DERV(UE(I-3),UE(I-2),UE(I-1),UE(I),UE(I+1),DUEDX,DX,4)
      GO TO 1270
 1260 CONTINUE
      CALL DERV(DODI(I-4),DODI(I-3),DODI(I-2),DODI(I-1),DODI(I),DDELDX,D
     1X,5)
      CALL DERV(UE(I-4),UE(I-3),UE(I-2),UE(I-1),UE(I),DUEDX,DX,5)
 1270 CONTINUE
      IF(NPR .LE. 4)GO TO 1320
      XN=XSTAR(I)/DODI(I)*DDELDX
 1300 CONTINUE
      IF(XN .LT. 0.) GO TO 1320
      IF(ABS(XN) .LE. .05) GO TO 1320
      IF(XN .LT. .99) GO TO 1340
      IF(ABS(XN-1.) .LE. .01) GO TO 1320
      GO TO 1350
 1320 CONTINUE
      EXPK=0.
      AYESL=AIEE(I)
      GO TO 1370
 1340 CONTINUE
      THETSL=(XN-1.)*THETAS(I)/RADIAN
      XK=2./3.*XN*(XN-1.)
      EXPK=-.194*EXP(-XK)
      GO TO 1360
 1350 CONTINUE
      THETSL=(1.-1./XN)*THETAS(I)/RADIAN
      XK=2./3.*(XN-1.)
      EXPK=.194*EXP(-XK)
 1360 CONTINUE
      VP=UE(I)*COS(THETSL)
      AYESL=H-.5*VP**2
```

```
 1370 CONTINUE
C
C        EQUIVALENT DISTANCE PARAMETERS
C
 1500 CONTINUE
C
      CALL EBAR(AYEW,AYESL,H,SIGR,EXPK,POPSLX(I),OMEGAE(I),OMEGAS,GAMC,
     1GAMO,EBARL,EBART)
      G=ROMUR*UE(I)
      FOFI=DODI(I)/RORI(I)
      GLAM=G*RORI(I)**2*FOFI**(2.*EBARL)
      GTURB=G*RORI(I)**1.25*FOFI**(1.25*EBART)
      IF(GSAVE-1 )1510,1520,1530
 1510 CONTINUE
      GSAVE=1
      SUML=GLAM*XI*BEQXL
      SUMT=GTURB*XI*BEQXT
      SL(1)=SUML
      ST(1)=SUMT
      SAVEL=SUML
      GO TO 1540
 1520 CONTINUE
      GSAVE=2
      GL2=GLAM
      GT2=GTURB
      GO TO 1001
 1530 CONTINUE
      GSAVE=1
      SUML=SUML+DX/3.*(GL1+4.*GL2+GLAM)
      SUMT=SUMT+DX/3.*(GT1+4.*GT2+GTURB)
      SL(I)=SUML
      ST(I)=SUMT
      BEQXL=SUML/(GLAM*XSTAR(I))
      BEQXT=SUMT/(GTURB*XSTAR(I))
      CX=1.-SAVEL/SUML
      FDX=71.*CX/(70.*CX+1.)
      BETAS=BETAS+FDX*(DBS-BETAS)
      SAVEL=SUML
 1540 CONTINUE
      IF(UE(I))1560,1550,1560
 1550 CONTINUE
      DBS=2.*(H/AIEE(I))*BEQXL
```

```
      GO TO 1570
 1560 CONTINUE
      DBS=2.*H/AIEE(I)*XSTAR(I)/UE(I)*DUEDX*BEQXL
 1570 CONTINUE
      GL1=GLAM
      GT1=GTURB
 1580 CONTINUE
C
C        BOUNDARY LAYER TRANSITION CHECK
      IF(RTRANS)1600,1583,1583
 1583 CONTINUE
      IF(ITRAN)1585,1585,1600
 1585 CONTINUE
      CALL JAYELL(AYEW(I),H,AIEE(I),ZTEX(I),ZTS,SIGR,BETAS,
     1POPSLX(I),XJL)
      CALL MUZERO(OMEGAR,ZTR,AYER,AIEE(I),H,POPSLX(I),XMUO,XL)
      SEQXL=BEQXL/XJL**2
      SEQXT=BEQXT/XJL**.5625
      SEQL=SEQXL*XSTAR(I)
      SEQT=SEQXT*XSTAR(I)
      XEQXL=BEQXL/XJL**.4
      XEQL=XEQXL*XSTAR(I)
      FXQ=CBRT(XJL**.85*BEQXT/BEQXL)
      RRQ=ROMUR*UE(I)*XEQL/XMUO**2/FXQ**2
      IF(RRQ-RTRANS)1600,1590,1590
 1590 CONTINUE
      IF(UE(I))1600,1600,1591
 1591 IF(XSTAR(I))1600,1600,1592
 1592 ITRAN=1
      RUM=ROMUR*UE(I)/XMUO**2
      RLS=RUM*SEQL
      RTS=RUM*SEQT
      XMUE=OMEGAE(I)*ZTEX(I)
      RTHL=XMUO/XMUE*.664*SQRT(RLS)
      ADEM=ALOG10(RTS)-.407
      IF(ADEM)1593,1593,1594
 1593 RTHT=0.
      CTR=0.
      GO TO 1595
 1594 RTHT=XMUO/XMUE*.2135*RTS/ADEM**2.64
      CTR=SUMT*(1.-(RTHL/RTHT)**1.25)
```

```
 1595 BEQXT=BEQXT-CTR/GTURB/XSTAR(1)
      XTRANS=XSTAR(1)
      ITR=1
      GO TO 1510
C        CHECK FOR PRINTOUT VALUES
 1600 CONTINUE
      IF(XSTAR(1)-XPO(K))1610,1620,1630
 1610 CONTINUE
      IF(XSTAR(1)+2.*DX-XPO(K))1001,1001,1630
 1620 CONTINUE
      IPRINT=1
 1630 CONTINUE
      CALL JAYELL(AYEW(I),H,AIEE(1),ZTEX(I),ZTS,SIGR,BETAS,
     1POPSLX(I),XJL)
      CALL MUZERO(OMEGAR,ZTR,AYER,AIEE(I),H,POPSLX(I),XMUO,XL)
      SEQXL=BEQXL/XJL**2
      XEQXL=BEQXL/XJL**.4
      SEQL=SEQXL*XSTAR(I)
      XEQL=XEQXL*XSTAR(I)
      SEQXT=BEQXT/XJL**.5625
      SEQT=SEQXT*XSTAR(I)
      BTBL=BEQXT/BEQXL
      FXS=(XJL**4.25*BTBL)**(1./3.)
      FXQ=(XJL**.85*BTBL)**(1./3.)
      RUM=ROMUR*UE(I)/XMUO**2
      RRS=RUM*SEQL/FXS**2
      RRQ=RUM*XEQL/FXQ**2
      P=XJL**1.5
      Q=XJL**.3
      FPM=FXS*P*XMUO
      TUEL=.332*FPM*SQRT(RRS)/SEQL
      TUET=FPM/SEQL*(.185*RRS/ALOG10(RRS+3000.)**2.584)
      RHOE=1.292*POPSLX(I)/ZTEX(I)
      XMUE=OMEGAE(I)*ZTEX(I)
      RLS=RUM*SEQL
      RTS=RUM*SEQT
      RTHL=XMUO/XMUE*.664*SQRT(RLS)
      IF(I .EQ. 1) GO TO 1633
      ADEM=ALOG10(RTS)-.407
      IF(ADEM)1631,1631,1632
 1631 RTHT=0.
      GO TO 1633
```

```
1632 RTHT=XMUO/XMUE*.2135*RTS/ADEM**2.64
1633 THETAL=XMUE/RHOE/UE(I)*RTHL
     THETAT=XMUE/RHOE/UE(I)*RTHT
     CFEL=2.*TUEL/RHOE/UE(I)
     CFET=2.*TUET/RHOE/UE(I)
     HL=.332*Q/SIGR**.645*XMUO*FXQ*SQRT(RRQ)/XEQL
     HL=ACCG*XL*HL
     IF(ITRAN .GE. 1) HL=0.
     HT=XMUO*FXQ*Q/SIGR**.645/XEQL*(.185*RRQ/ALOG10(RRQ+
    13000.)**2.584)
     HT=ACCG*XL*HT
     IF(ITRAN .LE. 0) HT=0.
     AYEAWL=H-(1.-SQRT(SIGR))*(H-AIEE(I))
     AYEAWT=H-(1.-CBRT(SIGR))*(H-AIEE(I))
     IF(I .LE. 1) GO TO 1660
     IF(ITC .LE. 2) GO TO 1660
     DO 1635 J=2,I,2
     SL(J)=(SL(J-1)+SL(J+1))/2.
     ST(J)=(ST(J-1)+ST(J+1))/2.
1635 CONTINUE
     PHISL=0.
     PHIST=0.
     AA=0
     DO 1640 J=2,I
     SLBS=(SL(J)-(SL(J)-SL(J-1))/3.)/SL(I)
     STBS=(ST(J)-(ST(J)-ST(J-1))/3.)/ST(I)
     FSBSL=1./CBRT(1.-SLBS**.75)
     STBS9=1.-STBS**.9
     IF(STBS9)1636,1639,1639
1636 IF(AA)1639,1637,1639
1637 WRITE(6,1638)
1638 FORMAT(1H0,5X,97H IN THE CALCULATION OF THE NON-ISOTHERMAL WALL PA
    1RAMETER  STBS9 A NEGATIVE VALUE WAS ENCOUNTERED.//6X,23HABS(STBS9)
    2WAS ASSUMED. )
     AA=1
1639 FSBST= ABS(STBS9)**(-1./9.)
     PHISL=(AYEW(J)-AYEW(J-1))*FSBSL+PHISL
     PHIST=(AYEW(J)-AYEW(J-1))*FSBST+PHIST
1640 CONTINUE
     PHIL=AYEW(I)-AYEW(1)-PHISL
     PHIT=AYEW(I)-AYEW(1)-PHIST
```

```
 1650 CONTINUE
      QDOTL=HL*(AYEAWL+PHIL-AYEW(I))/25031.3
      QDOTT=HT*(AYEAWT+PHIT-AYEW(I))/25031.3
      QLISO=HL*(AYEAWL-AYEW(I))/25031.3
      QTISO=HT*(AYEAWT-AYEW(I))/25031.3
      HL=HL/HO
      HT=HT/HREFT
      IF(ITR)1680,1680,1670
 1670 CONTINUE
      ITR=0
      GO TO 1600
 1680 CONTINUE
C
C     THIS COMPLETES THE CALCULATIONS FOR OBTAINING THE
C     HEAT TRANSFER COEFFICIENT AND SKIN FRICTION COEFFICIENT
C     FOR BOTH LAMINAR AND TURBULENT FLOW
C
C
C     OBTAINING THE ABOVE QUANTITIES AT THE REQUESTED
C     PRINT OUT POINTS
C
 1700 CONTINUE
      IF(INTERP)1701,1735,1701
C     INTERPOLATION NEEDED FOR PRINTOUTS
 1701 CONTINUE
      AYAWLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYAWLS,AYEAWL)
      AYAWTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYAWTS,AYEAWT)
      BQLINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),BEQXLS,BEQXL)
      BQTINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),BEQXTS,BEQXT)
      CFELT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),CFELS,CFEL)
      CFETT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),CFETS,CFET)
      DODINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),DODI(I-2),DODI(I))
      EBARLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),EBARLS,EBARL)
      EBARTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),EBARTS,EBART)
      ENTH=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AIEE(I-2),AIEE(I))
      FXQINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),FXQSAV,FXQ)
      FXSINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),FXSSAV,FXS)
      HLINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),HLSAVE,HL)
      HTINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),HTSAVE,HT)
      XJLNT =FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XJLSV,XJL)
      XNINT =FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XNSV,XN)
      OMEGNT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),OMEGAE(I-2),OMEGAE(I))
```

```
      OMEGRT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),OMEGRS,OMEGAR)
      PPSL=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),POPSLX(I-2),POPSLX(I))
      QDOTLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QDOTLS,QDOTL)
      QDOTTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QDOTTS,QDOTT)
      QLISOT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QLISOS,QLISO)
      QTISOT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QTISOS,QTISO)
      RORINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RORI(I-2),RORI(I))
      RRQINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RRQSAV,RRQ)
      RRSINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RRSSAV,RRS)
      THETLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETLS,THETAL)
      THETTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETTS,THETAT)
      TH8SNT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETAS(I-2),THETAS(I))
      TWINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),TW(I-2),TW(I))
      UEDGE=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),UE(I-2),UE(I))
      WENTH=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYEW(I-2),AYEW(I))
      XEQXLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XEQXLS,XEQXL)
      XEQXTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XEQXTS,XEQXT)
      XMUONT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XMUOSV,XMUO)
      ZTEINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),ZTEX(I-2),ZTEX(I))
      ZTRINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),ZTRSAV,ZTR)
      COUNT=COUNT+1.
      IF(COUNT-9.)1720,1720,1702
 1702 CONTINUE
      WRITE(6,6200)
      COUNT=1.
 1720 CONTINUE
      XSIPO=TBLP(X,XSIX,XPO(K),IP,MXSV)
      ETAPO=TBLP(X,ETAX,XPO(K),IP,NXSV)
      WRITE(6,6400) XPO(K),PPSL,UEDGE,DODINT,RORINT,TWINT,ZTEINT,OMEGNT,
     1ENTH,WENTH,
     2XSIPO,BQLINT,XEQXLT,AYAWLT,XJLNT,EBARLT,HLINT,QLISOT,QDOTLT,CFELT,
     3ETAPO,BQTINT,XEQXTT,AYAWTT,XNINT,EBARTT,HTINT,QTISOT,QDOTTT,CFETT,
     4TH8SNT,ZTRINT,XMUONT,FXQINT,FXSINT,RRQINT,RRSINT
     5,OMEGRT,THETLT,THETTT
 6400 FORMAT(1H0,4(10E12.4/))
      K=K+1
      INTERP=0
      IF(K-KF)1730,1730,1000
 1730 CONTINUE
      IF(XSTAR(I)-XPO(K))1760,1740,1701
C
```

```
C      PRINT OUT AND SAVE FOR INTERPOLATION
C
 1735 CONTINUE
      IF(IPRINT)1740,1765,1740
C      PRINT OUT CURRENT VALUES
 1740 CONTINUE
      COUNT=COUNT+1.
      IF(COUNT-9.)1750,1750,1742
 1742 CONTINUE
      WRITE(6,6200)
      COUNT=1.
 1750 CONTINUE
      WRITE(6,6400) XPO(K),POPSLX(I),UE(I),DODI(I),RORI(I),TW(I),ZTEX(I)
     1,OMEGAE(I),AIEE(I),AYEW(I),
     2XSIX(I),BEQXL,XEQXL,AYEAWL,XJL,EBARL,HL,QLISO,QDOTL,CFEL,
     3ETAX(I),BEQXT,XEQXT,AYEAWT,XN ,EBART,HT,QTISO,QDOTT,CFET,
     4THETAS(I),ZTR,XMUO,FXG,FXS,RRQ,RRS
     5,OMEGAR,THETAL,THETAT
      K=K+1
      IPRINT=0
      IF(K-KF)1760,1760,1000
 1760 CONTINUE
      IF(XSTAR(I)+2.*DX-XPO(K))1001,1001,1765
C      SAVE ALL QUANTITIES AT LEFT END OF INTERVAL
 1765 CONTINUE
      AYAWLS=AYEAWL
      AYAWTS=AYEAWT
      BEQXLS=BEQXL
      BEQXTS=BEQXT
      CFELS=CFEL
      CFETS=CFET
      EBARLS=EBARL
      EBARTS=EBART
      FXQSAV=FXQ
      FXSSAV=FXS
      HLSAVE = HL
      HTSAVE=HT
      XJLSV=XJL
      XNSV=XN
      OMEGRS=OMEGAR
      QDOTLS=QDOTL
      QDOTTS=QDOTT
```

```
      QLISOS=QLISO
      QTISOS=QTISO
      RRQSAV=RRQ
      RRSSAV=RRS
      THETLS=THETAL
      THETTS=THETAT
      XEQXLS=XEQXL
      XEQXTS=XEQXT
      XMUOSV=XMUO
      ZTRSAV=ZTR
      INTERP=1
      GO TO 1001
 1000 CONTINUE
      WRITE(6,6100)XTRANS
 6100 FORMAT(1H0,26HTRANSITION OCCURRED AT X =  ,E12.5)
      RETURN
      END
$IBFTC BEQI    DECK
      SUBROUTINE BEQI
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A(     5),BEQXL  ) , (A(     6),BEQXT  ) , (A(    11),DX     ),
     2(A(    15),H      ) , (A(    34),XI     ) , (A(   237),POPSLX),
     3(A(   987),DODI   ) , (A(  1737),RORI   ) , (A(  2737),AIEE   ),
     4(A(  3487),OMEGAE) , (A(  7737),XSTAR  ) , (A(  9237),THETAS),
     5(A(  9987),UE     ) , (A(10737),TW     ) , (IA(    8),NPR    )
      DIMENSION XSTAR(750),POPSLX(750),UE(750),AIEE(750),
     1OMEGAE(750),THETAS(750),DODI(750),RORI(750),TW(750)
      DIMENSION T(11),P(11),OMEG(11),DOD(11),ROR(11),X(11),
     1TH8S(11),U(11),FOF(11),AE(11)
C
      RADIAN=57.3
      BEQXLI=1.0
      BEQXTI=1.0
      T(1)=TW(1)
      P(1)=POPSLX(1)
      OMEG(1)=OMEGAE(1)
      DOD(1)=DODI(1)
      ROR(1)=RORI(1)
      X(1)=XSTAR(1)
      TH8S(1)=THETAS(1)
```

```
      U(1)=UE(1)
      AE(1)=AIEE(1)
      DO 300 ITER=1,5
      DX=DX/10.
      BX=1./10.**ITER
      DO 100 I=2,11
      B=I-1
      B=BX*B
      X(I)=X(1)+B*(XSTAR(2)-XSTAR(1))
      P(I)=P(1)+B*(POPSLX(2)-POPSLX(1))
      T(I)=T(1)+B*(TW(2)-TW(1))
      U(I)=U(1)+B*(UE(2)-UE(1))
      AE(I)=AE(1)+B*(AIEE(2)-AIEE(1))
      OMEG(I)=OMEG(1)+B*(OMEGAE(2)-OMEGAE(1))
      DOD(I)=DOD(1)+B*(DODI(2)-DODI(1))
      ROR(I)=ROR(1)+B*(RORI(2)-RORI(1))
      TH8S(I)=TH8S(1)+B*(THETAS(2)-THETAS(1))
  100 CONTINUE
C
      NG=0
      DO 200 I=1,11
      CALL IWALL(T(I),P(I),AYEW)
      CALL SOMEGA(H,P(I),ZTS,OMEGAS)
      CALL SOMEGA(AYEW,P(I),ZTW,OMEGAW)
      CALL RORMUR(P(I),OMEGAS,OMEG(I),OMEGAW,OMEGAR,ROMUR,ZTR,AYER,
     1SIGR)
C        DELTA DERIVATIVE
 1200 CONTINUE
      IF(I-2)1210,1220,1230
 1210 CONTINUE
      CALL DERV(DOD(I),DOD(I+1),DOD(I+2),DOD(I+3),DOD(I+4),DDELDX,DX,1)
      GO TO 1270
 1220 CONTINUE
      CALL DERV(DOD(I-1),DOD(I),DOD(I+1),DOD(I+2),DOD(I+3),DDELDX,DX,2)
      GO TO 1270
 1230 CONTINUE
      IF(I-10   )1240,1250,1260
 1240 CONTINUE
      CALL DERV(DOD(I-2),DOD(I-1),0.,DOD(I+1),DOD(I+2),DDELDX,DX,3)
      GO TO 1270
 1250 CONTINUE
      CALL DERV(DOD(I-3),DOD(I-2),DOD(I-1),DOD(I),DOD(I+1),DDELDX,DX,4)
```

```
      GO TO 1270
 1260 CONTINUE
      CALL DERV(DOD(I-4),DOD(I-3),DOD(I-2),DOD(I-1),DOD(I),DDELDX,DX,5)
 1270 CONTINUE
      IF(NPR .LE. 4) GO TO 1320
      XN=X(I)/DOD(I)*DDELDX
 1300 CONTINUE
      IF(XN .LT. 0.) GO TO 1320
      IF(ABS(XN) .LE. .05) GO TO 1320
      IF(XN .LT. .99) GO TO 1340
      IF(ABS(XN-1.) .LE. .01) GO TO 1320
      GO TO 1350
 1320 CONTINUE
      EXPK=0.
      AYESL=AE(I)
      GO TO 1370
 1340 CONTINUE
      THETSL=(XN-1.)*TH8S(I)
      XK=2./3.*XN*(XN-1.)
      EXPK=-.194*EXP(-XK)
      GO TO 1360
 1350 CONTINUE
      THETSL=(1.-1./XN)*TH8S(I)
      XK=2./3.*(XN-1.)
      EXPK=.194*EXP(-XK)
 1360 CONTINUE
      VP= U(I)*COS(THETSL/RADIAN)
      AYESL=H-.5*VP**2
 1370 CONTINUE
C
C        EQUIVALENT DISTANCE PARAMETERS
C
      CALL EBAR(AYEW,AYESL,H,SIGR,EXPK,P(I),OMEG(I),OMEGAS,GAMC,GAMO,
     1EBARL,EBART)
      G=ROMUR*U(I)
      FOF(I)=DOD(I)/ROR(I)
      GLAM=G*ROR(I)**2*FOF(I)**(2.*EBARL)
      GTURB=G*ROR(I)**1.2*FOF(I)**(1.2*EBART)
      IF(NG-1)151,152,153
  151 CONTINUE
      NG=1
```

```
      SUML=GLAM*XI*BEQXLI
      SUMT=GTURB*XI*BEQXTI
      GO TO 160
  152 CONTINUE
      NG=2
      GL2=GLAM
      GT2=GTURB
      GO TO 200
  153 CONTINUE
      NG=1
      SUML=SUML+DX/3.*(GL1+4.*GL2+GLAM)
      SUMT=SUMT+DX/3.*(GT1+4.*GT2+GTURB)
      BEQXL=SUML/(GLAM*X(I))
      BEQXT=SUMT/(GTURB*X(I))
  160 CONTINUE
      GL1=GLAM
      GT1=GTURB
  200 CONTINUE
      BEQXLI=BEQXL
      BEQXTI=BEQXT
  300 CONTINUE
      BEQXL=BEQXLI
      BEQXT=BEQXTI
C
  310 CONTINUE
      WRITE(6,6010)BEQXLI,BEQXTI
 6010 FORMAT(1H0,18HINITIAL BEQ VALUES/10H BEQXLI = ,E12.5/10H BEQXTI =
     1,E12.5//)
C
      DX=1.E+5*DX
      RETURN
      END
$IBFTC INTIAL  DECK
      SUBROUTINE INTIAL(L1,NTYPE)
C        CALCULATES INITIAL CONDITIONS
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A(    1),ACH    ) , (A(    3),ALT    ) , (A(   15),H      ),
     2(A(   18),PINF   ) , (A(   19),POPINF) , (A(   20),POPSL  ),
     3(A(   21),QINF   ) , (A(   25),RHO    ) , (A(   26),TEMP   ),
     4(A(   31),VEL    ) , (A(   32),VELP   )
```

```
C
C
C        WIND TUNNEL I
      GO TO(10,20,30),L1
   10 CONTINUE
      VEL=49.*ACH*SQRT(TEMP)
      H=6000.*TEMP+.5*VEL**2
      RHO=PINF/(1716.*TEMP)
      GO TO 40
C
C        WIND TUNNEL II
   20 CONTINUE
      AYEI=H-.5*VEL**2
      IF(AYEI-5930000.)21,22,22
   21 CONTINUE
      SONIC=.6325*SQRT(AYEI)
      GO TO 23
   22 CONTINUE
      CALL SONENT(AYEI,SONIC)
   23 CONTINUE
      ACH=VEL/SONIC
      TEMP=(VEL/49./ACH)**2
      RHO=PINF/(1716.*TEMP)
      GO TO 50
C
C        FLIGHT
   30 CONTINUE
      CALL ATMOS(ALT,SONIC,PINF,RHO,TEMP)
      H=6000.*TEMP+.5*VEL**2
      ACH=VEL/SONIC
   40 CONTINUE
      QINF=.7*PINF*ACH**2
   50 CONTINUE
C
      ACHN=ACH
      CALL STACON(PSTPNF,AYEST,VELP,ACH,ACHN,H,PINF,VEL)
      GO TO(60,70,60),L1
C
   60 CONTINUE
      POPINF=PSTPNF
      POPSL=POPINF*PINF/PSL
```

```
      GO TO 80
   70 CONTINUE
      POPINF=PSTPNF
      PINFP=POPSL*PSL/POPINF
      QINFP=PINFP*VEL**2/2./R/TEMP
      PINF=PINFP
      QINF=QINFP
   80 CONTINUE
C
      WRITE(6,6020)NTYPE,VEL,ACH,H,TEMP,PINF,QINF,POPSL,POPINF,ALT
 6020 FORMAT(1H0,23HINITIAL CONDITIONS FOR ,A6,5H CASE//4X,8HVELOCITY,4X
     1,4HMACH,8X,8HENTHALPY,4X,4HTINF,8X,4HPINF,8X,4HQINF,8X,6HPO/PSL,6X
     2,7HPO/PINF,4X,8HALTITUDE//1X,9E12.5///)
      RETURN
      END
$IBFTC REF      DECK
      SUBROUTINE REF
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
     1(A(     1),ACH    ) , (A(     4),AYEWO ) , (A(    15),H      ),
     2(A(    16),HO     ) , (A(    17),HREFT ) , (A(    18),PINF   ),
     3(A(    20),POPSL  ) , (A(    21),QINF  ) , (A(    23),RCLREF),
     4(A(    24),RHMREF) , (A(    29),TWREF ) , (A(    31),VEL    ),
     5(A(    32),VELP   )
C
C
      AYEE=H
      BETAS=0.5
C
      CALL IWALL(TWREF,POPSL,AYEW)
      CALL SOMEGA(AYEW,POPSL,ZTW,OMEGW)
      CALL SOMEGA(AYEE,POPSL,ZTE,OMEGE)
      CALL SOMEGA(H,POPSL,ZTS,OMEGS)
      CALL RORMUR(POPSL,OMEGS,OMEGE,OMEGW,OMEGR,ROMUR,ZTR,AYER,SIGR)
      CALL JAYELL(AYEW,H,AYEE,ZTE,ZTS,SIGR,BETAS,POPSL,XJL)
      CALL MUZERO(OMEGR,ZTR,AYER,AYEE,H,POPSL,XMUO,XL)
      HO=.664*ACCG*XL/SIGR**.645*SQRT(ROMUR*VEL*XJL*
     1VELP/RHMREF)
      QDOTO=HO*(H-AYEW)/25031.3
      AYEWO=AYEW
      ACHN=0.5*ACH
```

```
      EXPK=0.
      CALL STACON(PSTPNF,AYEST,VELP,ACH,ACHN,H,PINF,VEL)
      PSTPSL=PSTPNF*PINF/PSL
      CALL IWALL(TWREF,PSTPSL,AYEW)
      AYEE=AYEST
      AYESL=AYEST
      CALL SOMEGA(AYEW,PSTPSL,ZTW,OMEGW)
      CALL SOMEGA(AYEE,PSTPSL,ZTE,OMEGE)
      CALL SOMEGA(H,PSTPSL,ZTS,OMEGS)
      CALL RORMUR(PSTPSL,OMEGS,OMEGE,OMEGW,OMEGR,ROMUR,ZTR,
     1AYER,SIGR)
      CALL EBAR(AYEW,AYESL,H,SIGR,EXPK,PSTPSL,OMEGE,OMEGS,GAMC,GAMO,
     1EBARL,EBART)
      CALL MUZERO(OMEGR,ZTR,AYER,AYEE,H,PSTPSL,XMUO,XL)
      XEQL=.866*RCLREF/(1.+GAMC)/VELP
      FXQ=(1.6*(1.+GAMO)/(1.+.7652*GAMO))**.333*(1.+GAMO)/(1.+GAMC)
      RRQREF=.866*ROMUR*VEL*XEQL/(XMUO*FXQ)**2
      HREFL=.332*ACCG*XL/SIGR**.645*XMUO*FXQ/XEQL*SQRT(RRQREF)
      HREFT=.185*ACCG*XL/SIGR**.645*XMUO*FXQ/XEQL*RRQREF/
     1(ALOG10(RRQREF+3000.))**2.584
      AYAWRL=H-(1.-SQRT(SIGR))*(H-AYEE)
      AYAWRT=H-(1.-SIGR**.333)*(H-AYEE)
      QREFL=HREFL*(AYAWRL-AYEW)/25031.3
      QREFT=HREFT*(AYAWRT-AYEW)/25031.3
      TUREFL=SIGR**.645/ACCG/XL*HREFL
      TUREFT=SIGR**.645/ACCG/XL*HREFT
      TAURL=.866*VEL*TUREFL
      TAURT=.866*VEL*TUREFT
      CFREFL=TAURL/QINF
      CFREFT=TAURT/QINF
      AYEREF=AYEW
      HO24=.24*HO
      HRL24=.24*HREFL
      HRT24=.24*HREFT
      WRITE(6,6000) HO,HREFL,HREFT,AYAWRL,TAURL,TAURT,RHMREF,
     1HO24,HRL24,HRT24,AYAWRT,CFREFL,CFREFT,RCLREF,
     2QDOTO,QREFL,QREFT,AYEREF,AYEWO,TWREF
 6000 FORMAT(1HO,2HHO,10X,6HHREF,L,6X,6HHREF,T,6X,9HIAW,REF,L,3X,8HTAURE
     1F,L,4X,8HTAUREF,T,4X,7HREM,REF/
     26H .24HO,7X,9H.24HREF,L,3X,9H.24HREF,T,3X,9HIAW,REF,T,3X,11HCFINF,
     3REF,L,1X,11HCFINE,REF,T,9H RCYL,REF/
```

```
      46H QDOTO,7X,10HQDOT,REF,L,2X,10HQDOT,REF,T,2X,6HIW,REF,6X,4HIW,O,8
      5X,6HTW,REF//
      63(7E12.4/)//)
       RETURN
       END
$IBFTC SONENT  DECK
       SUBROUTINE SONENT(ENTH,SONIC,IERROR)
C      ROUTINE TO OBTAIN FREE STREAM SPEED OF SOUND
C      ENTHALPY IN FT**2/SEC**2 AND SPEED OF SOUND IN FT/SEC
       DIMENSION SONICT(10),AYE(10),LA(8)
       INTEGER XLOC
       DATA KSS/1/
       DATA(AYE(I),I=1,9)/
      15.93E6,8.475E6,10.17E6,11.86E6,13.56E6,15.26E6,16.95E6,
      218.64E6,20.34E6/
       DATA(SONICT(I),I=1,9)/
      11540.,1794.,1941.,2076.,2200.,2313.,2421.,2519.,2609./
       IF(ENTH .GT. 20.34E6)GO TO 30
       GO TO(10,20),KSS
    10 CONTINUE
       LA(1)=XLOC(LA(1))
       LA(2)=XLOC(AYE(1))
       LA(3)=XLOC(SONICT(1))
       LA(4)=1
       LA(5)=1
       LA(6)=1
       LA(7)=10
       KSS=2
    20 LA(8)=0
       SONIC=TAB(ENTH,LA(1))
       GO TO 40
    30 CONTINUE
       WRITE(6,6100)ENTH
  6100 FORMAT(1H0,63HFREE STREAM ENTHALPY EXCEEDS TABLE VALUE IN SONENT.
      1ENTHALPY = ,E12.5/25H MAXIMUM VALUE = 20.34E6. )
    40 CONTINUE
       RETURN
        END
$IBFTC STACON  DECK
       SUBROUTINE STACON(PSTPNF,AYEST,VELP,ACH,ACHN,H,PINF,VEL)
       DIMENSION L(8),ACOST(11),CORRT(11)
       DATA (ACOST(I),I=1,11)/.1,.2,.3,.5,.7,.8,1.0,1.2,1.5,2.0,2.5/
```

```
      DATA (CORRT(I),I=1,11)/1.122,1.118,1.1115,1.0955,1.068,1.055,
     11.033,1.021,1.0076,1.002,1.000/
      COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
      DATA (L(I),I=4,7)/ 1,1,2,11/
      AYEST=H-.5*VEL**2*(1.-(ACHN/ACH)**2)
      IF(ACHN-1.)10,20,20
   10 CONTINUE
      PSTO=PINF/PSL*(1.+ACHN**2/5.)**3.5
      GO TO 30
   20 CONTINUE
      PSTO=PINF/PSL*(1.2*ACHN**2)**3.5*(6./(7.*ACHN**2-1.))**2.5
   30 CONTINUE
      R2R10=6.*ACHN**2/(ACHN**2+5.)
      CALL SOMEGA(AYEST,PSTO,ZTSTO,OMEG)
      PST1=PINF/PSL+(PSTO-PINF/PSL)*((2.*R2R10-6006.*ZTSTO/AYEST)/
     1(2.*R2R10-1.))
      CALL SOMEGA(AYEST,PST1,ZTST1,OMEG)
      PST2=PINF/PSL+(PSTO-PINF/PSL)*((2.*R2R10-6006.*ZTST1/AYEST)/
     1(2.*R2R10-1.))
      PSTPNF=PST2*PSL/PINF
      VELPO=SQRT(2./7.*(1.+5./ACHN**2)*(1.-1./PSTPNF)*!6006.*
     1ZTST1/AYEST))
      IF(ACHN-2.5)40,40,50
C
   40 CONTINUE
      L(1)=LOC(L(1))
      L(2)=LOC(ACOST(1))
      L(3)=LOC(CORRT(1))
      L(8)=0
      CORR=TAB(ACHN,L(1))
      GO TO 60
C
   50 CONTINUE
      CORR=1.0
   60 CONTINUE
      VELP=VELPO*CORR**5
      RETURN
      END
$IBFTC ATMOS   DECK
      SUBROUTINE ATMOS(ALT,CS,P,DENS,T)
C     SAVE SENSE LIGHT
```

```
      CALL SLITET(1,K000FX)
       GO TO(4600,4700),K000FX
 4600 SAVE = 1.0
      GO TO 4800
 4700 SAVE = 0.0
 4800 CONTINUE
C     SET ERROR INDICATOR
      I = 0
C     CONVERT TO METERS
C
      Z = ALT*0.3048
C
C     COMPUTE GEOPOTENTIAL ALTITUDE
C
      IF (Z-90000.0) 1000, 1000, 2000
C     ALTITUDE LESS THAN NINETY THOUSAND METERS
 1000 Z = 6356766.0*Z/(6356766.0+Z)
      WEIMOL = 28.9644
      IF (Z) 1, 12, 3
    1 I = -1
      Z = 0.0
    2 GO TO 12
    3 IF (Z-79000.0) 5, 4, 4
    4 TMB = 180.65
      GRAD = 0.0
      ZB = 79000.0
      PB = 1.0377E-2
      GO TO 3000
    5 IF (Z-47000.0) 6, 7, 7
    6 IF (Z-20000.0) 8, 9, 9
    7 IF (Z-61000.0) 10, 11, 11
    8 IF (Z-11000.0) 12, 13, 13
    9 IF (Z-32000.0) 14, 15, 15
   10 IF (Z-52000.0) 16, 17, 17
   11 TMB = 252.65
      GRAD = -4.0
      ZB = 61000.0
      PB = 1.82099E-1
      GO TO 3000
   12 TMB = 288.15
      GRAD = -6.5
      ZB = 0.0
```

```
      PB = 1.01325E3                                              24192740
      GO TO 3000                                                  24192741
   13 TMB = 216.65                                                24192742
      GRAD = 0.0                                                  24192743
      ZB = 11000.0                                                24192744
      PB = 2.26320E2                                              24192745
      GO TO 3000                                                  24192746
   14 TMB = 216.65                                                24192747
      GRAD = 1.0                                                  24192748
      ZB = 20000.0                                                24192749
      PB = 5.47487E1                                              24192750
      GO TO 3000                                                  24192751
   15 TMB = 228.65                                                24192752
      GRAD = 2.8                                                  24192753
      ZB = 32000.0                                                24192754
      PB = 8.68014                                                24192755
      GO TO 3000                                                  24192756
   16 TMB = 270.65                                                24192757
      GRAD = 0.0                                                  24192758
      ZB = 47000.0                                                24192759
      PB = 1.10905                                                24192760
      GO TO 3000                                                  24192761
   17 TMB = 270.65                                                24192762
      GRAD = -2.0                                                 24192763
      ZB = 52000.0                                                24192764
      PB = 5.90005E-1                                             24192765
      GO TO 3000                                                  24192766
 2000 IF (Z-700000.0) 20, 20, 18                                  24192767
   18 I = 1                                                       24192768
      Z = 700000.0                                                24192769
   20 IF (Z-170000.0) 21, 22, 22                                  24192770
   21 IF (Z-110000.0) 23, 24, 24                                  24192771
   22 IF (Z-400000.0) 25, 26, 26                                  24192772
   23 IF (Z-100000.0) 27, 28, 28                                  24192773
   24 IF (Z-150000.0) 29, 30, 30                                  24192774
   25 IF (Z-230000.0) 31, 32, 32                                  24192775
   26 IF (Z-500000.0) 33, 34, 34                                  24192776
   27 TMB = 180.65                                                24192777
      GRAD = 0.003                                                24192778
      ZB = 90000.0                                                24192779
      PB = 0.16437801                                             24192780
```

```
      GO TO 4000                                                        24192781
   28 TMB = 210.65                                                      24192782
      GRAD = 0.005                                                      24192783
      ZB = 100000.0                                                     24192784
      PB = .30074613E-01                                                24192785
      GO TO 4000                                                        24192786
   29 IF (Z-120000.0) 35, 36, 36                                        24192787
   30 IF (Z-160000.0) 37, 38, 38                                        24192788
   31 IF (Z-190000.0) 39, 40, 40                                        24192789
   32 IF (Z-300000.0) 41, 42, 42                                        24192790
   33 TMB = 2160.65                                                     24192791
      ZB = 400000.0                                                     24192792
      PB = .40303676E-05                                                24192793
      GRAD = 0.0026                                                     24192794
      GO TO 4000                                                        24192795
   34 IF (Z-600000.0) 43, 44, 44                                        24192796
   35 TMB = 260.65                                                      24192797
      ZB = 110000.0                                                     24192798
      PB = .73544163E-02                                                24192799
      GRAD = 0.01                                                       24192800
      GO TO 4000                                                        24192801
   36 TMB = 360.65                                                      24192802
      GRAD = 0.02                                                       24192803
      ZB = 120000.0                                                     24192804
      PB = .25216515E-02                                                24192805
      GO TO 4000                                                        24192806
   37 TMB = 960.65                                                      24192807
      GRAD = 0.015                                                      24192808
      ZB = 150000.0                                                     24192809
      PB = .50617014E-03                                                24192810
      GO TO 4000                                                        24192811
   38 TMB = 1110.65                                                     24192812
      GRAD = 0.01                                                       24192813
      ZB = 160000.0                                                     24192814
      PB = .36942892E-03                                                24192815
      GO TO 4000                                                        24192816
   39 TMB = 1210.65                                                     24192817
      GRAD = 0.007                                                      24192818
      ZB = 170000.0                                                     24192819
      PB = .27925979E-03                                                24192820
      GO TO 4000                                                        24192821
   40 TMB = 1350.65                                                     24192822
```

```
      GRAD = 0.005                                                      24192823
      ZB = 190000.0                                                     24192824
      PB = .16852213E-03                                                24192825
      GO TO 4000                                                        24192826
   41 TMB = 1550.65                                                     24192827
      GRAD = 0.004                                                      24192828
      ZB = 230000.0                                                     24192829
      PB = .69604207E-04                                                24192830
      GO TO 4000                                                        24192831
   42 TMB = 1830.65                                                     24192832
      GRAD = 0.0033                                                     24192833
      ZB = 300000.0                                                     24192834
      PB = .18838460E-04                                                24192835
      GO TO 4000                                                        24192836
   43 TMB = 2420.65                                                     24192837
      GRAD = 0.0017                                                     24192838
      ZB = 500000.0                                                     24192839
      PB = .10956796E-05                                                24192840
      GO TO 4000                                                        24192841
   44 TMB = 2590.65                                                     24192842
      GRAD = 0.0011                                                     24192843
      ZB = 600000.0                                                     24192844
      PB = .34502145E-06                                                24192845
      GO TO 4000                                                        24192846
 3000 TM = TMB+GRAD*(Z-ZB)*0.001                                        24192847
      T = TM                                                            24192848
      CS = SQRT(1.4*TM/0.00348368)                                      24192849
      GO TO 5000                                                        24192850
 4000 TM = TMB+GRAD*(Z-ZB)                                              24192851
      IF (Z-170000.0) 4400, 4500, 4500                                  24192852
 4400 CONTINUE                                                          24192853
C     Z BETWEEN 90000.M AND 170000.M,COMPUTE M                          24192854
      EM =(((((.141865090E-27*Z -.111458341E-21) *Z +.359201416E-16)*Z -24192855
     1.606652130E-11) *Z +.565215183E-06)*Z -.275300356E-01) *Z +5.769524192856
     2191E2                                                             24192857
      WEIMOL = EM*28.9644/28.966                                        24192858
      GO TO 4050                                                        24192859
 4500 CONTINUE                                                          24192860
C     Z GREATER THAN 170000.M,COMPUTE M                                 24192861
      EM   =(((((1.1805540E-33*Z   -3.4291620E-27)                      24192662
     1*Z   +3.8159669E-21)*Z   -2.0402054E-15)                          24192863
```

```
     2*Z    +5.6422141E-10)*Z    -1.0700489E-4)
     3*Z    +3.5629995E1
      WEIMOL = EM*28.9644/28.966
 4050 CONTINUE
C     COMPUTE TEMPERATURE
      T = TM*WEIMOL/28.9644
C     SPEED OF SOUND IS CONSTANT
      CS = 269.44
C     COMPUTE PRESSURE
      EMOR = 0.00348368
      C = (-ZB+TMB/GRAD)*0.000001
      D=6.3675708
      CMD=C-D
      CMD2=CMD*CMD
      CMD3=CMD2*CMD
      CMD4=CMD2*CMD2
      CMD5=CMD2*CMD3
      CMD6=CMD3*CMD3
      CMD7=CMD4*CMD3
      CMD8=CMD4*CMD4
      CMD9=CMD5*CMD4
      CMD10=CMD5*CMD5
      C2=C*C
      C3=C2*C
      D2=D*D
      PBLN=ALOG(PB)
      TEM14 = EMOR/GRAD
      CALL SLITE (1)
C     COMPUTE USING ZB
      Z6=ZB*.000001
 4100 ZPD=Z6+D
      ZPC=Z6+C
      ZPD2=ZPD*ZPD
      ZPD3=ZPD2*ZPD
      ZPD4=ZPD2*ZPD2
      ZPD5=ZPD2*ZPD3
      ZPD6=ZPD3*ZPD3
      ZPD7=ZPD3*ZPD4
      ZPD8=ZPD4*ZPD4
      ZPD9=ZPD5*ZPD4
      ZPC2=ZPC*ZPC
      ZPC3=ZPC2*ZPC
```

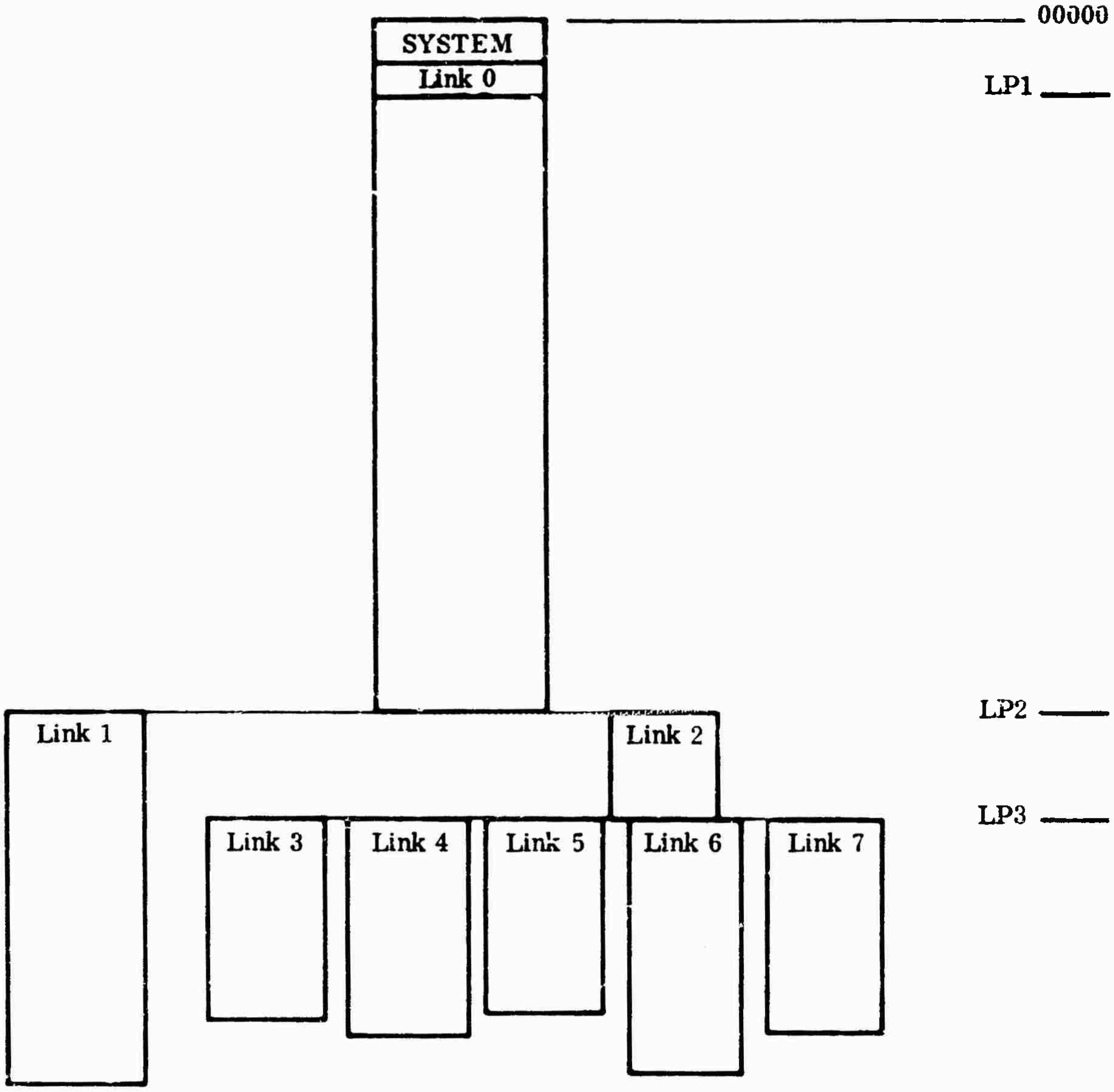
```
      ZDLOG=ALOG(ABS(ZPD))+6.*ALOG(10.)
      ZCLOG=ALOG(ABS(ZPC))+6.*ALOG(10.)
      ZCDLOG=ALOG(ABS(ZPC/ZPD))
      TEMP1=1./(-CMD*ZPD)+ZCDLOG/CMD2
      TEMP2=-ZDLOG/CMD4-1./(CMD3*ZPD)+1./(2.*CMD2*ZPD2)-1./(3.*CMD*ZPD3)
     1+ZCLOG/CMD4
      TEMP3= D*ZDLOG/CMD4-ZPD/CMD4+ZDLOG/CMD3+D/(ZPD*CMD3)
     1+1./(ZPD*CMD2)-D/(2.*ZPD2*CMD2)-1./(2.*ZPD2*CMD)
     2+D/(3.*ZPD3*CMD)+ZPC/CMD4-C*ZCLOG/CMD4
      TEMP4=-ZPD2/(2.*CMD4)+2.*D*ZPD/CMD4-D2*ZDLOG/CMD4
     1+ZPD/CMD3-2.*D*ZDLOG/CMD3-D2/(ZPD*CMD3)
     2-ZDLOG/CMD2-2.*D/(ZPD*CMD2)+D2/(2.*ZPD2*CMD2)
     3-1./(ZPD*CMD)+D/(ZPD2*CMD)-D2/(3.*ZPD3*CMD)
     4+ZPC2/(2.*CMD4)-2.*C*ZPC/CMD4+C2*ZCLOG/CMD4
      TEMP5=ZCDLOG/CMD4-1./(CMD3*ZPD)+1./(2.*ZPD2*CMD2)
     1-1./(3.*ZPD3*CMD)
      TEM6=ZPC-C*ZCLOG
      TEM7=ZPC2/2.-2.*C*ZPC+C2*ZCLOG
      TEM8=ZPC3/3.-3.*C*ZPC2/2.+3.*C2*ZPC-C3*ZCLOG
      TEM11=-ZDLOG/CMD10-1./(CMD9*ZPD)+1./(2.*CMD8*ZPD2)
     1-1./(3.*CMD7*ZPD3)+1./(4.*CMD6*ZPD4)-1./(5.*CMD5*ZPD5)
     2+1./(6.*CMD4*ZPD6)-1./(7.*CMD3*ZPD7)+1./(8.*CMD2*ZPD8)
     3-1./(9.*CMD*ZPD9)+ZCLOG/CMD10
      TEM12=ZDLOG/CMD5+1./(CMD4*ZPD)-1./(2.*CMD3*ZPD2)
     1+1./(3.*CMD2*ZPD3)-1./(4.*CMD*ZPD4)-ZCLOG/CMD5
      TEM13=398.626940*TEMP1-39.227164*TEMP2-.041699200*TEMP3 -1.4755045
     1E-7*TEMP4+26.327892*TEMP5-.26586747E-02*TEM6 +2.8074026E-6*TEM7+9.
     29336241E-12*TEM8-.017043125*ZCLOG +1.4388483E3*TEM11 +.29060526*TE
     3M12+1.4673439E-5*ZCLOG
      CALL SLITET(1,K000FX)
       GO TO(4200,4300),K000FX
 4200 TEM15=TEM13
C     REPEAT USING Z
      Z6=Z   *.000001
      GO TO 4100
 4300 P =EXP(PBLN-TEM14*(TEM13-TEM15))
C     CONVERT TO MB
      P = P*0.01
      GO TO 8000
 5000 IF (GRAD) 6000, 7000, 6000
 6000 P = PB*(TMB/TM)**(9.80665*28.9644/(8.31432*GRAD))
```

```
      GO TO 8000
 7000 P = PB*EXP(-9.80665*28.9644*0.001*(Z-ZB)/(8.31432*TMB))
 8000 CONTINUE
      DENS = 28.9644*P/(8.31432*TM)*0.1
C     CONVERT DENSITY(KG/M3) TO LB/FT3
      DENS = DENS*(0.3048)**3/0.45359237
C     CONVERT DENSITY TO LB(MASS)/FT3
      DENS = DENS/32.1741
C     CONVERT TO P(MM HG)
      P = P*.750061682E+00
C     CONVERT TO IN HG
      P = P*.393700787E-01
C     CONVERT TO LB/FT2
      P = P*70.7269
C     CONVERT CS(M/SEC2) TO CS(FT/SEC2)
      CS = CS*3.2808399
C     CONVERT T(K) TO T(R)
      T = T*1.8
C     RESTORE SENSE LIGHT
 8100 IF (SAVE) 8200, 8300, 8200
 8200 CALL SLITE (1)
      GO TO 8400
 8300 CALL SLITET(1,K000FX)
       GO TO(8400,8400),K000FX
 8400 CONTINUE
      IF (I) 300, 9000, 200
  200 WRITE (6,400)
      GO TO 9000
  300 WRITE (6,500)
 9000 CONTINUE
      RETURN
  400 FORMAT(1H0,////5X,79HALTITUDE INPUT TO ATMOS EXCEEDED ROUTINE RANG
     1E - ASSIGNED VALUE OF 2,300,000 FT////)
  500 FORMAT(1H0,////5X,68HALTITUDE INPUT TO ATMOS BELOW ROUTINE RANGE -
     1 ASSIGNED VALUE OF 0 FT////)
      END
```

002982

b. $\rho_r \mu_r$ Deck Setup

The following diagrams indicate the deck setup and the core storage requirements for the $\rho_r \mu_r$ program. Each link is composed of the subroutine decks (symbolic or object) listed in the allocation diagram. All control cards are indicated by a $ in column 1 and must be punched exactly as shown in the deck setup diagram. All items in parenthesis indicate subroutine or data card decks.



Notes:

1. Storage locations are on actual numbers.

2. Load points (LP) are indicated as follows:

   LP1 - 04076

   LP2 - 60005

   LP3 - 65623

SCHEMATIC DIAGRAM OF COMPUTER STORAGE ALLOCATION

JOB TITLE	FOR ORGN. NO.	ANALYST	DATE	PAGE
				01

P_r M_r DECK SETUP INSTRUCTIONS AS 2419 IBM 7094

```
$JOB
$*      ACCTTIME
$*      PRINT ESTIMATE
$EXECUTE       IBJOB          00000 77777
$IBJOB         MAP FIOCS
$ORIGIN        ALPHA
Link 0 deck
$ORIGIN        BETA
Link 1 deck
$ORIGIN        BETA
Link 2 deck
$ORIGIN        GAMMA
Link 3 deck
$ORIGIN        GAMMA
Link 4 deck
$ORIGIN        GAMMA
Link 5 deck
$ORIGIN        GAMMA
Link 6 deck
$ORIGIN        GAMMA
Link 7 deck
$DATA
Data deck
$EOF
$IBSY
```

The contents of each link are:

Link 0
- AS2419 (main program)
- TAB
- DTAB
- DBTP
- TBLP
- XTAB
- LOC
- XLOC

Link 1
- DIVERG
- FIND
- STREAM

Link 2
- WALLT 1
- WALLT 2
- BLKDTA
- DERV
- EBAR
- IWALL
- JAYELL
- MUZERO
- RORMUR
- SOMEGA
- TABLE 2
- TABLE 6
- TABLE 7
- TABLE 14
- TABLE 18
- TABLE 19
- TABLE 20

Link 3
- DELTA

Link 4
- CYLIND
- FLOW

Link 5
- AXI2D
- HEMI

Link 6
- QTRAN
- BEQI

Link 7
- INTIAL
- REF
- SONENT
- STACON
- ATMOS

SECTION III

THE TURBULENT NONSIMILAR BOUNDARY LAYER PROGRAM (NSBL)

1. NOMENCLATURE

Program Symbol	Math Symbol	Description	Units
CP	c_p	specific heat of air at constant pressure, 6006	ft^2/sec^2-°R
	δ	boundary layer thickness	ft
* DELETA	$\delta(\eta/\rho)$	increment in boundary layer parameter	ft^3/slug
DELSTR	δ^*	boundary layer displacement thickness	ft
* DHDX	$\partial I/\partial x$	derivative of total enthalpy	ft/sec^2
* DHDY	$\partial I/\partial y$	derivative of total enthalpy	ft/sec^2
* DMUDY	$\partial\mu/\partial y$	derivative of viscosity	$lb\text{-}sec/ft^3$
DQ	$\partial\dot{q}/\partial y$	derivative of the heating rate	Btu/ft^3-sec
DRDX	$\partial r/\partial x$	derivative of three-dimensional flow parameter	---
DS	ΔS	increment for stagnation region problems	ft
* DUDX	$\partial u/\partial x$	derivative of velocity	1/sec
* DUDY	$\partial u/\partial y$	derivative of velocity	1/sec
* DUDYL	$(\partial u/\partial y)_e$	velocity derivative at boundary layer edge	1/sec
* DVDY	$\partial v/\partial y$	derivative of velocity normal to surface	1/sec
DX	Δx	increment in x-direction	ft

* Those symbols marked with an asterisk are arrays (tables) in the program. Throughout this document, a subscript i on either the program symbol or the equivalent math symbol indicates that reference is being made to the i^{th} position within the array.

Program symbol	Math symbol	Description	Units
DY	Δy	increment in y-direction	ft
* D2HDY	$\partial^2 I/\partial y^2$	second derivative of total enthalpy	$1/sec^2$
* D2UDY	$\partial^2 u/\partial y^2$	second derivative of velocity	1/sec-ft
ENDX	x_f	final value of x	ft
* ENTH	i	static enthalpy	ft^2/sec^2
EPS	ϵ	eddy viscosity	$lb\text{-}sec/ft^2$
EPSY	$\partial\epsilon/\partial y$	derivative of eddy viscosity	$lb\text{-}sec/ft^3$
* ETA	η	similarity parameter	---
* H	I	total enthalpy	ft^2/sec^2
* PR	σ	Prandtl number	---
* PRESSURE	P	pressure	lb/ft^2
* QP	$\dot{q}$	heating rate	$Btu/ft^2\text{-}sec$
QW	$\dot{q}_w$	heating rate at the wall	$Btu/ft^2\text{-}sec$
QQW	QQW	heating rate at the wall	$Btu/ft^2\text{-}sec$
R	r	three-dimensional flow parameter	ft
	R	universal gas constant, 1716	$ft^2/sec^2\text{-}^\circ R$
* RHO	ρ	density	$lb\text{-}sec^2/ft^4$
SI	S_I	initial value of S for the calculation of η	ft
STARTX	x_o	initial value of x	ft
* T	T	temperature	°R
* TAU	τ	shear force	lb/ft^2
TAULP	τ_L	local laminar shear force	lb/ft^2

Program symbol	Math symbol	Description	Units
TAULW	$\tau_{w,L}$	laminar shear force at the wall	lb/ft^2
TAULY	$\partial\tau_L/\partial y$	derivative of the laminar shear force	lb/ft^3
TAUTP	τ_T	local turbulent shear force	lb/ft^2
TAUTW	$\tau_{w,T}$	turbulent shear force at the wall	lb/ft^2
TAUTY	$\partial\tau_T/\partial y$	derivative of the turbulent shear force	lb/ft^3
TAUW	τ_w	shear force at the wall	lb/ft^2
THETA	θ	boundary layer momentum thickness	ft
TURBS	$\dfrac{\partial\tau_T/\partial y}{\partial\tau_L/\partial y}$	ratio of the turbulent to the laminar shear force derivative	---
* U	u	velocity in the x-direction	ft/sec
* V	v	velocity, $\tilde{v}$ in Volume I (Appendix C)	ft/sec
X	x	coordinate tangent to the surface	ft
XI	x_I	initial value of x for the calculation of η	ft
* XMU	μ	absolute viscosity of air	$lb\text{-}sec/ft^2$
* Y	y	coordinate normal to the surface	ft

Subscript notation:

Subscript	
e	evaluated at the boundary layer edge
i	evaluated at the i^{th} position within an array
I	initial value
L	laminar
o	initial value
T	turbulent
w	evaluated at the surface

2. METHOD OF SOLUTION

a. Forward Integration

In the following discussion, the equation numbers in parenthesis refer to an appropriate equation in Appendix C of Volume I. The equations solved by this computer program are the equation of state (C-3),

$$\rho = P/(RT)$$

Continuity (C-17),

$$v_i = \frac{1}{\left[\frac{\partial u_i}{\partial y} - \frac{2u_i}{\Delta y}\right]} \left\{ u_i^2 \left(\frac{1}{P}\frac{\partial P}{\partial x} + \frac{1}{r}\frac{\partial r}{\partial x}\right)_i + \frac{1}{\rho_i}\left[-\frac{\partial P}{\partial x} + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) + \frac{\partial \tau_T}{\partial y}\right]_i \right.$$

$$\left. - \frac{u_i}{i\rho_i}\left[\frac{\partial}{\partial y}\left(\frac{\mu}{\sigma}\frac{\partial i}{\partial y}\right) + \frac{\partial \dot{q}_T}{\partial y} + \frac{\partial u}{\partial y}\left(\mu\frac{\partial u}{\partial y} + \tau_T\right) + u\frac{\partial P}{\partial x}\right]_i - \frac{2u_i\, v_{i-1}}{\Delta y} - \left(\frac{\partial v}{\partial y}\right)_{i-1} u_i \right\}$$

x-momentum (C-6),

$$\frac{\partial u}{\partial x} = \frac{1}{\rho u}\left[-\frac{\partial P}{\partial x} + \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y} + \tau_T\right)\right] - \frac{v}{u}\frac{\partial u}{\partial y}$$

and Energy (C-7):

$$\frac{\partial I}{\partial x} = \frac{1}{\rho u}\frac{\partial}{\partial y}\left[\frac{\mu}{\sigma}\frac{\partial i}{\partial y} + \dot{q}_T + u\left(\mu\frac{\partial u}{\partial y} + \tau_T\right)\right] - \frac{v}{u}\frac{\partial I}{\partial y}$$

In addition, the relationship for turbulent shear stress, Equation (C-39), is required for the computation.

$$\frac{\partial \tau_T}{\partial y} = \left(\frac{\rho}{\rho_e}\right)^4 \left[.054\left(\frac{\rho u y}{\mu}\right)^{.833}\left(\frac{u}{u_e}\right)^{12} \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right)\right]$$

$$\epsilon_i = \frac{\tau_T}{\left(\frac{\partial u}{\partial y}\right)}$$

$$\dot{q}_T = \frac{\epsilon}{\sigma}\frac{\partial i}{\partial y}$$

Since v is expressed as a function of input data it can be determined explicitly at each point in the boundary layer at the initial or start position.

The y derivatives in the above expressions are calculated using three-point central differences at all positions except at $y = \Delta y$. Since the turbulent velocity and enthalpy profiles change very rapidly near the wall, an off-center method of calculation of derivatives is used at $y = \Delta y$.

$$\left(\frac{\partial u}{\partial y}\right)_2 = \frac{1}{6\Delta y}(9u_3 - 2u_4 - 6u_2)$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right) = \frac{1}{6\Delta y^2}(u_4 - 3u_3 - 9u_2)$$

With v defined and with u, I and y derivatives, $\partial u/\partial x$ and $\partial I/\partial x$ can now be determined. With the $\partial u/\partial x$ and $\partial I/\partial x$ derivatives determined the u and I profiles at the next station $(x_o$[1] $+ \Delta x)$ can be obtained by forward integration using:

$$u_{x+\Delta x} = u_x + \left(\frac{\partial u}{\partial x}\right)_x \Delta x$$

$$I_{x+\Delta x} = I_x + \left(\frac{\partial I}{\partial x}\right)_x \Delta x$$

The y derivatives, v, $\partial u/\partial x$, and $\partial I/\partial x$ calculations move out from the wall along a line normal to the wall until a specified limit is reached. The profiles are then stepped forward and the calculations repeated.

At each point in the boundary layer, a similarity parameter η is calculated as indicated below in subsection b., item (4). A value of η_{max} is an item of input used to limit the calculation in the y-directions, which are to be printed out.

Also calculated at each station x are the displacement and momentum thickness, heating rate, and the laminar and turbulent shear stress at the wall using:

$$\delta^* = \int_0^\delta \left[1 - \frac{\rho u}{\rho_e u_e}\right] dy$$

$$\theta = \int_0^\delta \left[\frac{\rho u}{\rho_e u_e}\left(1 - \frac{u}{u_e}\right)\right] dy$$

---

1 The notation x_o will be used throughout this section to denote the initial value of x.

$$\dot{q}_w = -\frac{1}{778} \int_0^{\delta} \left[\rho u \frac{\partial I}{\partial y} + \rho v \frac{\partial I}{\partial y} - u\tau \right] dy$$

$$\tau_L = \int_0^{\delta} \left[\mu \frac{\partial^2 u}{\partial y^2} + \frac{\partial \mu}{\partial y} \frac{\partial u}{\partial y} \right] dy$$

$$\tau_T = \int_0^{\delta} \frac{\partial \tau_T}{\partial y} dy$$

The heating rate and shear at the wall are calculated by an integration over the boundary layer rather than from the definitions because of the greater accuracy which can be obtained with the equation.

b. Numerical Method of Solution

The purpose of this section is to give the procedure used in solving the partial differential equations defined previously including the required numerical approximations.

The following variables and tables are input. Tables are indicated as functions of some independent variable, e.g. P = f(x) indicates the table of pressure as a function of x.

$u = f(y)$ at x_o	$\frac{\partial r}{\partial x} = f(x)$
$I = f(y)$ at x_o	$\left.\frac{\partial u}{\partial y}\right\|_e = f(x)$
$P = f(x)$	$v_1 = f(x)$
$r = f(x)$	$i_w = f(x)$

In addition, the following quantities must be specified:

Δx Δy η_{max} x_o x_f s_I x_I

(1) Calculation of tabular values

$$\left.\frac{\partial P}{\partial x}\right|_i = \left[(P_{i+1} - P_i)\frac{x_i - x_{i-1}}{x_{i+1} - x_i} + (P_i - P_{i-1})\frac{x_{i+1} - x_i}{x_i - x_{i-1}}\right]\left[\frac{1}{x_{i+1} - x_{i-1}}\right]$$

$$i = 2,\ 3,\ \ldots,\ (NP-1)$$

$$\left.\frac{\partial P}{\partial x}\right|_{i=1} = 0$$

(2) Values obtained from tables (linear interpolation)

$$P = f(x) \qquad \left(\frac{\partial u}{\partial y}\right)_e = f(x)$$

$$\frac{\partial P}{\partial x} = f(x) \qquad v_1 = f(x)$$

$$r = f(x) \qquad i_w = f(x)$$

$$\frac{\partial r}{\partial x} = f(x)$$

(3) Computation at initial $x = x_o$

$$y_i = (i - 1)\,\Delta y \qquad i = 1,\ 2,\ 3,\ \ldots,\ 200$$

$$u_i = u_{i-1} + \left(\frac{\partial u}{\partial y}\right)_e \Delta y \qquad i = (LIST + 1),\ \ldots,\ 200$$

$$I_i = I_{LIST}$$

LIST = number of input values in u and I tables.

$$T_i = (I - u^2/2)/C_P$$

$$\mu_i = (2.272 \times 10^{-8}\ T_i^{3/2})/(T_i + 198.6)$$

$$\rho_i = P_i/(1716\ T_i)$$

$$C_P = 6006\ \mathrm{ft}^2/\mathrm{sec}^2\text{-}^\circ R$$

$$i = 2,\ 3,\ \ldots,\ 200.$$

(4) Calculation of $\delta\eta$

$$\delta\left(\frac{\eta}{\rho}\right) = (u_e\, r\, \Delta y\, \mu_e^{3/5})/S^{4/5}$$

$$S = S_I + \rho_e u_e \mu_e r^2 x_I \text{ initially. At } x + \Delta x$$

$$S_{x+\Delta x} = \rho_e u_e \mu_e r^2 \Delta x$$

(5) Calculation of y-derivatives and v-profile

$$\left.\frac{\partial v}{\partial y}\right|_1 = \frac{v_1\left(\frac{T_2}{T_1} - 1\right)}{\Delta y}$$

$$\left.\frac{\partial \mu}{\partial y}\right|_i = \frac{\mu_{i+1} - \mu_{i-1}}{2\,\Delta y}$$

$$\left.\begin{aligned}
\left.\frac{\partial u}{\partial y}\right|_2 &= \frac{1}{6\Delta y}(9u_3 - 2u_4 - 6u_2) \\
\left.\frac{\partial^2 u}{\partial y^2}\right|_2 &= \frac{1}{6\Delta y^2}(u_4 - 3u_3 - 9u_2) \\
\left.\frac{\partial I}{\partial y}\right|_2 &= \frac{1}{6\Delta y}(9I_3 - 2I_4 - 6I_2 - I_1) \\
\left.\frac{\partial^2 I}{\partial y^2}\right|_2 &= \frac{1}{6\Delta y^2}(I_4 - 3I_3 - 9I_2 + 5I_1)
\end{aligned}\right\}\quad i = 2$$

$$\left.\frac{\partial u}{\partial y}\right|_i = \frac{u_{i+1} - u_{i-1}}{2\Delta y}$$

$$\left.\frac{\partial I}{\partial y}\right|_i = \frac{I_{i+1} - I_{i-1}}{2\Delta y}$$

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_i = \frac{u_{i+1} + u_{i-1} - 2u_i}{\Delta y^2}$$

$$\left.\frac{\partial^2 I}{\partial y^2}\right|_i = \frac{I_{i+1} + I_{i-1} - 2 I_i}{\Delta y^2}$$

$$\left.\frac{\partial \tau_L}{\partial y}\right|_i = \mu_i \left.\frac{\partial^2 u}{\partial y^2}\right|_i + \left.\frac{\partial \mu}{\partial y}\right|_i \left.\frac{\partial u}{\partial y}\right|_i$$

$$\left.\frac{\partial \tau_T}{\partial y}\right|_i = .054 \left(\frac{\rho_i u_i y_i}{\mu_i}\right)^{.833} \left(\frac{u_i}{u_e}\right)^{12} \left(\frac{T_e}{T_i}\right)^4 \left(\frac{\partial \tau_L}{\partial y}\right)_i$$

$$\tau'_L = \tau'_L\Big|_{i-1} + \left[\left(\frac{\partial \tau_L}{\partial y}\right)_{i-1} + \left(\frac{\partial \tau_L}{\partial y}\right)_i\right]\frac{\Delta y}{2}$$

$$\tau'_T = \tau'_{T_{i-1}} + \left[\left(\frac{\partial \tau_T}{\partial y}\right)_{i-1} + \left(\frac{\partial \tau}{\partial y}\right)_i\right]\frac{\Delta y}{2}$$

$$\eta_i = \eta_{i-1} + (\rho_i + \rho_{i-1})\ \delta\left(\frac{\eta}{\rho}\right)\frac{1}{2}$$

$$i = 2,\ 3,\ \ldots,\ \mathrm{MIN}\ (j,\ 199)$$

The edge of the boundary layer is defined by $j = i$ at which the following criteria is satisfied:

$$\frac{\Delta y}{u}\left[\frac{\partial u}{\partial y} - \left.\frac{\partial u}{\partial y}\right|_e\right] \le .00001 \qquad \frac{\Delta y}{I}\frac{\partial I}{\partial y} \le .00005$$

The following are calculated for the range $\_ \le i \le j$

$$\tau_{T_i} = \tau'_T\Big|_i - \tau'_T\Big|_j$$

$$\tau_{L_i} = \tau'_L\Big|_i - \tau'_L\Big|_j$$

$$\tau_i = \tau_{L_i} - \tau_{T_i}$$

$$\tau_1 = - \tau'_T\Big|_j - \tau'_L\Big|_j$$

$$\epsilon_i = \tau_{T_i} / (\partial u / \partial y)_i$$

$$\left.\frac{\partial \epsilon}{\partial y}\right|_i = \left\{ .054 \left(\frac{\rho_i u_i y_i}{\mu_i} \right)^{.833} \left(\frac{u_i}{u_e} \right)^{12} \left(\frac{T_e}{T_i} \right)^{.4} \left.\frac{\partial \tau_L}{\partial y}\right|_i - \epsilon_i \left.\frac{\partial^2 u}{\partial y^2}\right|_i \right\} \left\{ \frac{1}{(\partial u/\partial y)_i} \right\}$$

$$\left.\frac{d\dot{q}}{dy}\right|_i = \frac{1}{\sigma} \left\{ (\mu_i + \epsilon_i) \left.\frac{\partial^2 I}{\partial y^2}\right|_i + \left(\left.\frac{\partial \mu}{\partial y}\right|_i + \left.\frac{\partial \epsilon}{\partial y}\right|_i \right) \left.\frac{\partial I}{\partial y}\right|_i \right.$$

$$\left. - \frac{1}{2\Delta y} (u_{i+1}\ \tau_{i+1} - u_{i-1}\ \tau_{i-1}) \right\}$$

$$\dot{q}_i = \dot{q}_{i-1} + \left(\left.\frac{d\dot{q}}{dy}\right|_{i-1} + \left.\frac{\partial \dot{q}}{\partial y}\right|_i \right) \frac{\Delta y}{2}$$

$$\text{PART } 1_i = u_i^2 \left(\frac{1}{r} \frac{dr}{dx} + \frac{1}{P} \frac{dP}{dx} \right)$$

$$\text{PART } 2_i = \frac{1}{\rho_i u_i} \left(\left.\frac{\partial \tau_L}{\partial y}\right|_i + \left.\frac{\partial \tau_T}{\partial y}\right|_i - \left.\frac{\partial P}{\partial x}\right|_i \right)$$

$$\text{PART } 3_i = \frac{1}{\rho_i u_i} \left[\left.\frac{d\dot{q}}{dy}\right|_i + (u_{i+1}\ \tau_{i+1} - u_{i-1}\ \tau_{i-1}) \right] \frac{1}{2\Delta y}$$

$$\text{PART } 6_i = u_i \left[\frac{2\, v_{i-1}}{\Delta y} + \left.\frac{\partial v}{\partial y}\right|_{i-1} \right]$$

$$v_i = \frac{1}{\left.\frac{\partial u}{\partial y}\right|_i - \frac{2 u_i}{\Delta y}} \left[\text{PART } 1_i + u_i \text{ PART } 2_i - \frac{u_i^2}{(I - u_i^2/2)} (\text{PART } 3_i \right.$$

$$\left. - u_i \text{ PART } 2_i) - \text{PART } 6_i \right]$$

$$\left.\frac{\partial v}{\partial y}\right|_i = \frac{2}{\Delta y} (v_i - v_{i-1}) - \left.\frac{\partial v}{\partial y}\right|_{i-1}$$

(6) Calculation of x derivatives

$$\left.\frac{\partial u}{\partial x}\right|_i = \text{PART } 2_i - \frac{v_i}{u_i}\left.\frac{\partial u}{\partial y}\right|_i$$

$$\left.\frac{\partial I}{\partial x}\right|_i = \text{PART } 3_i - \frac{v_i}{u_i}\left.\frac{\partial I}{\partial y}\right|_i$$

(7) Calculation of δ^*, θ, Q_w, $\tau_{w,L}$, $\tau_{w,T}$ at x

$$\delta^* = \Delta y \sum_2^j \left\{1 - \frac{\rho_i u_i + \rho_{i-1} u_{i-1}}{2\rho_j u_j}\right\}$$

$$\theta = \frac{\Delta y}{2} \sum_2^j \left\{\frac{\rho_i u_i}{\rho_j u_j}\left(1 - \frac{u_i}{u_j}\right) + \frac{\rho_{i-1} u_{i-1}}{\rho_j u_j}\left(1 - \frac{u_{i-1}}{u_j}\right)\right\}$$

$$\dot{q}_w = \frac{-\Delta y}{778} \sum_2^j \left\{\frac{1}{2}\left[\rho_i u_i \left.\frac{\partial I}{\partial x}\right|_i + \rho_{i-1} u_{i-1} \left.\frac{\partial I}{\partial x}\right|_{i-1} + \rho_i v_i \left.\frac{\partial I}{\partial y}\right|_i + \rho_{i-1} v_{i-1} \left.\frac{\partial I}{\partial y}\right|_{i-1}\right] - (u_i \tau_i - u_{i-1} \tau_{i-1}) \frac{1}{\Delta y}\right\}$$

$$QQ_w = \frac{1}{778}\left\{\frac{-\dot{q}_j + \dfrac{\tau_1^2}{\mu_i}\dfrac{\Delta y}{2}}{1 + \dfrac{\rho_1 v_1 \sigma}{\mu_1}\dfrac{\Delta y}{2}}\right\}$$

$$\tau_{w,L} = -\tau'_{Lj}$$

$$\tau_{w,T} = -\tau'_{Tj}$$

(8) Calculation of u and I-profiles at $x + \Delta x$

$$\left.\begin{aligned} I_{x+\Delta x} &= I_x + \left.\frac{\partial I}{\partial x}\right|_x \Delta x \\ u_{x+\Delta x} &= u_x + \left.\frac{\partial u}{\partial x}\right|_x \Delta x \end{aligned}\right\} \quad i = 1, 2, \ldots, j$$

$$\left.\begin{aligned} u_i\Big|_{x+\Delta x} &= u_j\Big|_{x+\Delta x} \\ I_i\Big|_{x+\Delta x} &= I_j\Big|_{x+\Delta x} \end{aligned}\right\} \quad i = j+1,\ j+2,\ \ldots,\ 200$$

c. Stagnation Region Calculation

A slight modification to the previous discussed procedure is used to obtain stagnation point profiles. Input profiles are corrected by integrating the u and I profiles, but x (not equal to zero) is not increased during the integration. The profiles are assumed correct when

$$\left|\left(\frac{\partial u}{\partial x}\right)_i - \frac{u_i}{x}\right| < .1\, u_{LIST} \text{ for all } i$$

This convergence criteria is obtained with the velocity similarity stated below.

$$\left(\frac{\partial u_e}{\partial x}\right) = \frac{u_e}{u}\frac{\partial u}{\partial x}$$

$$\frac{\partial I_e}{\partial x} = 0$$

Experience has indicated that an enthalpy profile convergence criterion is not necessary.

For the stagnation region, the equations for the velocity and enthalpy profiles are changed to

$$I_{i,S} = I_{i,S-\Delta S} + \frac{\partial I}{\partial x}\Big|_i \Delta S$$

$$u_{i,S} = \left[u_{i,S-\Delta S} + \frac{\partial u}{\partial x}\Big|_i \Delta S\right]\frac{x_o}{x_o + \Delta S}$$

If I_i or u_i become negative for some i, ΔS is halved and the case is restarted. This may be repeated a maximum of two times for any combination of I_i or $u_i < 0$.

Additional input quantities are ΔS and MITR.

MITR is the maximum number of iterations that the program will make in trying to converge on a solution to the stagnation problem.

ΔS is a pseudo length similar to Δx. The program also uses ΔS as a test to determine whether the stagnation point calculation is to be performed or bypassed. For a flat plate case $\Delta S = 0$.

Calculations for a hemisphere may be performed near the stagnation point and also along the surface. This is accomplished by the input of $\Delta S > 0$ and $\Delta x > 0$.

3. INPUT-OUTPUT DESCRIPTION

a. Data Input Preparation

Data are input through the medium of IBM cards. The purpose of this section is to define the required inputs and the form they take on the cards.

A sample data sheet may be found at the end of this section. Use of this type of form greatly reduces the amount of effort required to obtain results from the non-similar boundary layer program. The data may be keypunched directly from the form.

All cards except cards 1 and 4 must be keypunched with decimal points and power of ten indicators as follows: if only a decimal point is needed, the number may be punched anywhere within the respective ten column field; however, if the power of ten indicator E is used, the exponent must be right-adjusted in the field.

The information on card 1 is alphanumeric and may be keypunched as desired.

The data on card 4 are all integers and are keypunched as fixed point numbers (i.e., no decimal or power of ten indicator E is to be used). Each number on card 4 must be right-adjusted in its respective five column field. For example, if JOT1 = 1 then a one must be keypunched in column five.

1) Data Card Formats

Card	Columns	Format	Description
1	1-48	8A6	Any information may be given on this card. It is used to identify the case.
2	1-10	7F10	Δx, the increment in the x direction. See the discussion on numerical stability in Appendix C (Volume I) for the criteria to be used in choosing a value.
2	11-20	7F10	Δy, the increment in the y direction. Should be chosen to provide at least ten points in the boundary layer for accurate results.

Card	Columns	Format	Description
2	21-30	7F10	η_{max}, the limiting value on the number of calculations in the y direction.
2	31-40	7F10	x_0, the initial value of x. $x_0 > 0$.
2	41-50	7F10	x_f, the final value of x.
2	51-60	7F10	σ, the turbulent Prandtl number.
2	61-70	7F10	CHECK, case dependence criteria. If CHECK = 0 then the current case is not dependent on previous calculations. If CHECK = 1 then the current case is dependent on the previous case.
3	1-10	4F10	ΔS. Calculations may be made for a blunt body. This involves calculating profiles near the stagnation point as well as along the body. This type of problem is indicated by $\Delta S > 0$ and MITR > 0 (card 4). For nonstagnation point calculations, $\Delta S = 0$.
3	11-20	4F10	Turbulent Prandtl number - must be equal to Prandtl number on card 2.
3	21-30	4F10	S_I, value of S used to initialize $S = S_I + \rho_e u_e r^2 x_I$
3	41-50	4F10	x_I, value of x used to initialize $S = S_I + \rho_e u_e r^2 x_I$

The data on card 4 control the number of values to be input to each of the succeeding tables. The values must be greater than zero. There is an upper limit on the number of values that may be input to each table. This limit is given with the description of the table. The limit may be changed by altering DIMENSION statements in the program.

Card	Columns	Format	Description
4	1-5	14I5	JOT1, the number of tabular values in the DJOT and XJOTT tables. $0 < \text{JOT1} \leq 10$.
4	6-10	14I5	LIST, the number of tabular values in the u and I profile tables. $2 \leq \text{LIST} \leq 50$.

Card	Columns	Format	Description
4	11-15	14I5	NR, the number of tabular values in the three-dimensional flow parameter table (r), the table of its x-derivatives, and the tables of their corresponding values of x. $2 \leq NR \leq 50$.
4	16-20	14I5	NP, the number of tabular values in the pressure table and the table of corresponding values of x. $2 \leq NP \leq 50$.
4	21-25	14I5	NW, the number of tabular values in the wall enthalpy table and the table of corresponding values of x. $2 \leq NW \leq 50$.
4	26-30	14I5	NDU, the number of tabular values in the $(\partial u/\partial y)_e$ table and the table of corresponding values of x. $2 \leq NDU \leq 50$.
4	31-35	14I5	NV1, the number of tabular values of velocity normal to the surface and the table of corresponding values of x. $2 \leq NV1 \leq 50$.
4	36-40	14I5	MITR, the maximum number of iterations allowed to calculate the profiles for the stagnation region.

Cards 5 and 6 are used to specify different frequencies of output and the corresponding ranges. The maximum number of values in these tables is 10.

Card	Columns	Format	Description
5[2]	-----	7F10	Table DJOT, assigns value to first, second, and successive printout intervals.
6	-----	7F10	Table XJOTT, assigns value of x up to which each interval of table DJOT is used.

2 In the case of tables with more than seven input values, the card number refers to a set of cards (e.g., if ten entries are made in the DJOT table, then card 5 actually refers to two cards).

Example: $x_o = 0.5$ and $x_f = 1.0$. Printout is desired every .05 (value of x) until x = .25, every .01 until x = .3, and every .1 until $x = x_f$. The required input would be:

Card	Column	Value	
4	1-5	3	(right-adjusted)
5	1-10 11-20 21-30	.05 .01 .1	
6	1-10 11-20 21-30	.25 .3 1.0	

Card	Columns	Format	Description
7	-----	7F10	Table u-profile = f(y). The purpose of this table is to provide the program with an initial velocity profile at $x = x_o$. Normally, this table must be input and contain at least three values. Exception is made when a case is dependent on a previous case (see Section 2). This table contains LIST tabular values. The maximum number of values in this table is 50.
7	1-10	7F10	Velocity at wall.
7	11-20	7F10	Velocity at Δy from wall.
7	21-30	7F10	Velocity at $2\Delta y$ from wall.
8	-----	7F10	Table I-profile = f(y). The purpose of this table is to provide the program with an initial total enthalpy profile at $x = x_o$. Normally, this table must be input and contain at least three values. Exception is made when a case is dependent on a previous case (see Section 2). This table must contain the same number of values (LIST) as the u-profile table. The maximum number of values in this table is 50. Entries are assumed to be consistent with Δy.
8	1-10	7F10	Total enthalpy at the wall.

Card	Columns	Format	Description
8	11-20	7F10	Total enthalpy at Δy from the wall.
8	21-30	7F10	Total enthalpy at $2\Delta y$ from the wall.
9	-----	7F10	Table $r = f(x)$. This table contains NR tabular values of r, the three-dimensional flow parameter (streamline divergence due to body geometry). The maximum number of values in this table is 50.
10	-----	7F10	Table $(\partial r/\partial x) = f(x)$. This table contains NR tabular values of $(\partial r/\partial x)$. The maximum number of values in this table is 50.
11	-----	7F10	Table x. This table contains NR tabular values of x corresponding to entries made in the r and $(\partial r/\partial x)$ tables (cards 9 and 10).
12	-----	7F10	Table pressure = f(x). This table contains NP tabular values of pressure. This table is numerically differentiated for $(\partial P/\partial x)$. The first value of $(\partial P/\partial x)$ is set to zero. The maximum number of values in this table is 50.
13	-----	7F10	Table x. This table contains NP tabular values of x corresponding to the entries made in the P table (card 12).
14	-----	7F10	Table wall enthalpy = f(x). This table contains NW tabular values of the total enthalpy at the wall. This table is used for nonisothermal wall problems. The maximum number of values in this table is 50.
15	-----	7F10	Table x. This table contains NW tabular values of x corresponding to the entries made in the wall enthalpy table (card 14).
16	-----	7F10	Table $(\partial u/\partial y)_e = f(x)$. This table contains NDU tabular values of $(\partial u/\partial x)$ evaluated at the edge of the boundary layer. For all problems not involving vorticity $(\partial u/\partial x) = 0$. The maximum number of values in this table is 50.

Card	Columns	Format	Description
17	-----	7F10	Table x. This table contains NDU tabular values of x corresponding to the entries made in the $(\partial u/\partial y)_e$ table (card 16).
18	-----	7F10	Table $V_w = f(x)$. This table contains NV1 tabular values of the velocity normal to the wall. This table is used for mass injection or leakage problems. The maximum number of values in this table is 50.
19	-----	7F10	Table x. This table contains NV1 tabular values of x corresponding to the entries made in the V_w table (card 18).
20			NOTE: In all cases where the values in the above tables are zero, a zero must be entered on the input sheet.

Example: the normal velocity at the wall is zero:

V_w table (card 14)	0.0	0.0	0.0
x table (card 15)	0.0	1.0	2.0

2) Case Dependence

The NSBL program has a criterion which allows a case to be dependent on the success or failure of the previous case. This involves the input CHECK and LIST and the test for ERROR.

If:			Then:
LIST	CHECK	ERROR	
0	0	---	Not dependent on the previous case, program continues.
0	1	0	Dependent on the previous case which worked, program continues, using final profile from previous case.
0	1	1	Dependent on the previous case which failed, so present case stops and program continues to next case.
0	---	0	Dependent on the previous case which worked, program continues, using final profile from previous case.
0	---	1	Dependent on the previous case which failed, so present case stops and program continues to next case.

b. Output Description

The printed output consists of the input data and at each printout interval the following are printed as a function of y:

y	u	$\dot{q}$	$\partial v/\partial y$	v	$\partial u/\partial x$
T	τ	I	η	$\partial I/\partial x$	ϵ

The following are printed as a function of the x at the end of each printout interval:

$\dot{q}_w$

$\tau_{w,L}$

$\tau_{w,T}$

δ^*

θ

ρ_1

$\partial P/\partial x$

QQW

S

c. NSBL Input Form

CARD

CARD							
1	TITLE CARD	FORMAT 8A6	ALPHANUMERIC ANY SPACE 1-48				
2	Δx	Δy	η_{max}	x_o	x_f	Prandtl no.	CHECK
3	ΔS	Prandtl no.	S_I	x_I	50	60	70

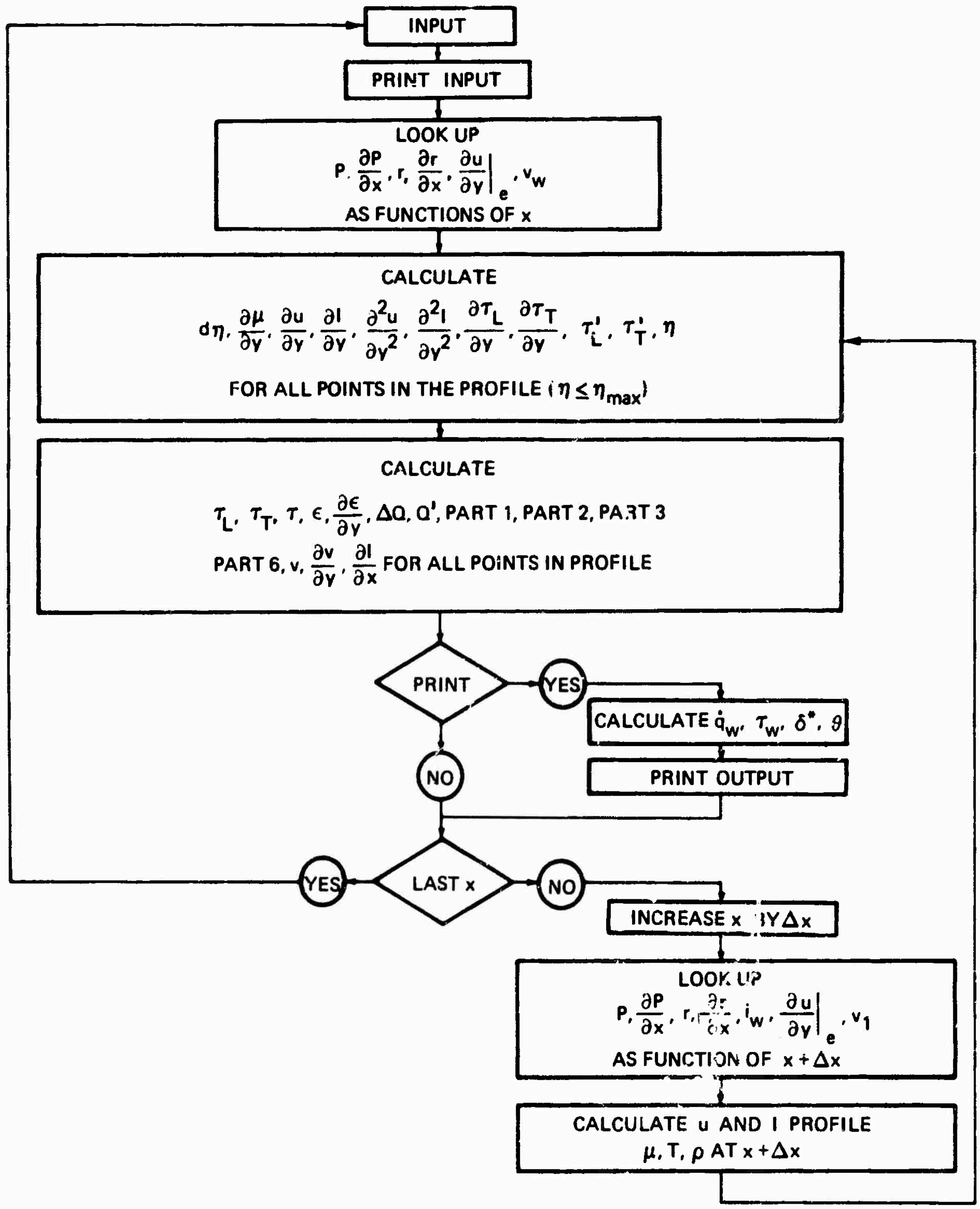
CARD								
4	JOT1	LIST	NR	NP	NW	NDU	NV1	MITR
	5	10	15	20	25	30	35	40

CARD	Label
5	DJOT
6	XJOTT
7	u-profile
8	l-profile
9	r
10	dr/dx
11	x table
12	Pressure
13	x table
14	Wall enthalpy
15	x table
16	$(du/dy)_e$
17	x table
18	v_w
19	x table

1 11 21 31 41 51 61 71

4. PROGRAMMING INFORMATION

a. Program Flow Chart



b. NSBL Program Listing

```
$IBFTCTA1781T  DECK
C      NON-SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS
       DIMENSION Y(200),U(200),T(200),V(200),DUDY(200),D2UDY(200),DHDY(20
      10),D2HDY(200),DHDX(200),ETA(200),H(200),PT(50),XT(50),DPT(50),LDP(
      28),LP(8),LV1(8),XV1T(50),V1T(50),RHO(200),XMU(200),DMUDY(200),
      3WALLH(50),XWH(50),LH(8),DJOT(10),XJOTT(10),DUDX(500),
      4DVDY(500),LR(8),RT(50),XRT(50),LDR(8),DRT(50),TITLE(8),LDU(8),DUT(
      550),XDUT(50),USVE(200),HSVE(200),TSVE(200),XMUSVE(200),RHOSVE(200)
      6,        TAU(200),TURBS(200)
       DIMENSION TAULY(200),TAUTY(200),TAULP(200),TAUTP(200),
      1TAUT(200),TAUL(200),EPS(200),EPSY(200),DQ(200),QP(200),Q(200)
       INTEGER XLOC
    1  FORMAT(7F10.0)
    2  FORMAT(29H PRESSURE TABLE LOOK-UP ERROR)
    5  FORMAT(25H DPDX TABLE LOOK-UP ERROR)
    6  FORMAT(34H WALL-ENTHALPY TABLE LOOK-UP ERROR)
    7  FORMAT(14I5)
    8  FORMAT(15H0PRESSURE TABLE/19X,1HX,12X,8HPRESSURE)
    9  FORMAT(20H0WALL-ENTHALPY TABLE/19X,1HX,7X,13HWALL-ENTHALPY)
   10  FORMAT(50H1NON-SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS    ,10X,
      18A6,/,3H0X=F8.5)
   11  FORMAT(1H0,8X,1HY,19X,1HU,19X,1HQ,18X,4HDVDY,17X,1HV,18X,4HDUDX/9X
      1,1HT,18X,3HTAU,18X,1HH,18X,3HETA,16X,4HDHDX,17X,3HEPS//)
   12  FORMAT(2(6E20.6/))
   13  FORMAT(27H SOME T(I) IS LESS THAN 0.0)
   14  FORMAT(31H SOME U(I) IS LESS THAN (-U(J)))
   15  FORMAT(39H SOME H(I) IS LESS THAN OR EQUAL TO 0.0)
   16  FORMAT(1H0,11HTRY NUMBER I1,30H WILL BE MADE USING DX EQUALS F8.6/
      1/)
   17  FORMAT(2F20.8)
   19  FORMAT(31H R FUNCTION TABLE LOOK-UP ERROR)
  200  FORMAT(25H DRDX TABLE LOOK-UP ERROR)
  201  FORMAT(50H1NON-SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS    ,10X,
      18A6)
  202  FORMAT( 34H0INPUT DATA FOR THE FOLLOWING CASE,/,77H (ANY QUANTITIE
      1S NOT SHOWN HERE RETAIN THE VALUES USED IN THE PRECEDING CASE))
  203  FORMAT(15H0CASE CONSTANTS,/,5X,3HDX=E12.5,5X,3HDY=E12.5,6X,7HETAMA
      1X=F7.3,/,1X,7HSTARTX=E12.5,3X,5HENDX=E12.5,10X,3HPR=F7.4)
  204  FORMAT(13H PRINT EVERY ,E12.5,9H UNTIL X=E12.5)
  205  FORMAT(8H0R TABLE/19X,1HX,19X,1HR,15X,5HDR/DX)
  206  FORMAT(28H0Q-WALL FOR THIS PROFILE IS E12.5/39H TAU-WALL (LAMINAR)
```

```
   1 FOR THIS PROFILE IS E12.5/41H TAU-WALL (TURBULENT) FOR THIS PROFI1781T042
   2LE IS E12.5/32H DELTA STAR FOR THIS PROFILE IS E12.5/             1781T043
   226H THE MOMENTUM THICKNESS IS E12.5/                      47H THE1781T044
   2CONVERGENCE CRITERION WAS SATISFIED AFTER ,I3,9H PROFILES)        1781T045
 207 FORMAT(1H0,26X,65H*** THIS CASE WAS NOT RUN DUE TO DEPENDENCE ON P1781T046
   1RECEDING CASE ***)                                                1781T047
 208 FORMAT(3F20.5)                                                   1781T048
 209 FORMAT(1H0,11HDU/DY TABLE/19X,1HX,15X,5HDU/DY)                   1781T049
 210 FORMAT(1H ,24HDUDY TABLE LOOK-UP ERROR)                          1781T050
 211 FORMAT(8A6)                                                      1781T051
 216 FORMAT(1H ,24HV(1) TABLE LOOK UP ERROR)                          1781T052
 217 FORMAT(1H0,10HV(1) TABLE/19X,1HX,16X,4HV(1))                     1781T053
C                                                                     1781T054
C      PROGRAM CONSTANTS                                              1781T055
C                                                                     1781T056
       DJOT(1)=0.0                                                    1781T057
       XJOTT(1)=0.0                                                   1781T058
       CP = 6006.                                                     1781T059
       DMUDY(1)=0.0                                                   1781T060
       ETA(1)=0.                                                      1781T061
C                                                                     1781T062
C      TABF CALLING SEQUENCES                                         1781T063
C                                                                     1781T064
C      ***PRESSURE                                                    1781T065
C                                                                     1781T066
       LP(1)=XLOC(LP(1))                                              1781T067
       LP(2)=XLOC(XT(1))                                              1781T068
       LP(3)=XLOC(PT(1))                                              1781T069
       LP(4)=1                                                        1781T070
       LP(5)=1                                                        1781T071
       LP(6)=1                                                        1781T072
       LP(7)=50                                                       1781T073
C                                                                     1781T074
C      ***DPDX                                                        1781T075
C                                                                     1781T076
       LDP(1)=XLOC(LDP(1))                                            1781T077
       LDP(2)=XLOC(XT(1))                                             1781T078
       LDP(3)=XLOC(DPT(1))                                            1781T079
       LDP(4)=1                                                       1781T080
       LDP(5)=1                                                       1781T081
       LDP(6)=1                                                       1781T082
       LDP(7)=50                                                      1781T083
```

```
C                                                           1781T084
C     ***R FUNCTION                                         1781T085
C                                                           1781T086
      LR(1)=XLOC(LR(1))                                     1781T087
      LR(2)=XLOC(XRT(1))                                    1781T088
      LR(3)=XLOC(RT(1))                                     1781T089
      LR(4)=1                                               1781T090
      LR(5)=1                                               1781T091
      LR(6)=1                                               1781T092
      LR(7)=50                                              1781T093
C                                                           1781T094
C     ***DRDX                                               1781T095
C                                                           1781T096
      LDR(1)=XLOC(LDR(1))                                   1781T097
      LDR(2)=XLOC(XRT(1))                                   1781T098
      LDR(3)=XLOC(DRT(1))                                   1781T099
      LDR(4)=1                                              1781T100
      LDR(5)=1                                              1781T101
      LDR(6)=1                                              1781T102
      LDR(7)=50                                             1781T103
C                                                           1781T104
C     ***WALL ENTHALPY                                      1781T105
C                                                           1781T106
      LH(1)=XLOC(LH(1))                                     1781T107
      LH(2)=XLOC(XWH(1))                                    1781T108
      LH(3)=XLOC(WALLH(1))                                  1781T109
      LH(4)=1                                               1781T110
      LH(5)=1                                               1781T111
      LH(6)=1                                               1781T112
      LH(7)=50                                              1781T113
C                                                           1781T114
C     ***DUDX                                               1781T115
C                                                           1781T116
      LDU(1)=XLOC(LDU(1))                                   1781T117
      LDU(2)=XLOC(XDUT(1))                                  1781T118
      LDU(3)=XLOC(DUT(1))                                   1781T119
      LDU(4)=1                                              1781T120
      LDU(5)=1                                              1781T121
      LDU(6)=1                                              1781T122
      LDU(7)=50                                             1781T123
C                                                           1781T124
```

```
C        ***V(1)
C
         LV1(1)=XLOC(LV1(1))
         LV1(2)=XLOC(XV1T(1))
         LV1(3)=XLOC(V1T(1))
         LV1(4)=1
         LV1(5)=1
         LV1(6)=1
         LV1(7)=50
      18 READ (5,211)(TITLE(I),I=1,8)
         READ(5,1)DX,DY,ETAMAX,STARTX,ENDX,PR,CHECK,DS,PRT,SI,XI
         X=STARTX
         LENGTH=ENDX/DX+1.
         LX=1
         XJOT=STARTX
         TIMES=0.0
         NTRY=1
         READ (5,7)JOT1,LIST,NR,NP,NW,NDU,NV1,MITR
         KK=200
         LIST4=4*LIST
         IF(LIST)185,185,180
     180 J=LIST4
     185 JOT=JOT1+1
         READ (5,1)(DJOT(I),I=2,JOT)
         READ (5,1)(XJOTT(I),I=2,JOT)
         WRITE (6,201)(TITLE(I),I=1,8)
         WRITE (6,202)
         WRITE (6,203)DX,DY,ETAMAX,STARTX,ENDX,PR
         WRITE (6,204)(DJOT(I),XJOTT(I),I=2,JOT)
         IF(LIST)252,252,93
    93   READ (5,1)(U(I),I=1,LIST)
         READ (5,1)(H(I),I=1,LIST)
         IF(NR)81,81,80
C
C        R TABLE AND DR/DX TABLE
C
      80 READ (5,1)(RT(I),I=1,NR)
         READ (5,1)(DRT(I),I=1,NR)
         READ (5,1)(XRT(I),I=1,NR)
         WRITE (6,205)
         WRITE (6,208)(XRT(I),RT(I),DRT(I),I=1,NR)
      81 IF(NP)  94,94,96
```

```
C                                                                       1781T167
C     PRESSURE TABLE                                                    1781T168
C                                                                       1781T169
   96 READ (5,1)(PT(I),I=1,NP)                                          1781T170
      READ (5,1)(XT(I),I=1,NP)                                          1781T171
      WRITE (6,8)                                                       1781T172
      WRITE (6,17)(XT(I),PT(I),I=1,NP)                                  1781T173
      NP=NP-1                                                           1781T174
C                                                                       1781T175
C     CALCULATE DPDX TABLE                                              1781T176
C                                                                       1781T177
      DPT(1)=0.0                                                        1781T178
      DO 100 I=2,NP                                                     1781T179
      DEL1=XT(I+1)-XT(I)                                                1781T180
      DEL2=XT(I)-XT(I-1)                                                1781T181
      DEL3=DEL1+DEL2                                                    1781T182
      DPT(I)=((PT(I+1)-PT(I))*DEL2/DEL1+(PT(I)-PT(I-1))*DEL1/DEL2)/DEL3 1781T183
  100 CONTINUE                                                          1781T184
   94 IF(NW)98,98,97                                                    1781T185
C                                                                       1781T186
C     WALL ENTHALPY TABLE                                               1781T187
C                                                                       1781T188
   97 READ (5,1)(WALLH(I),I=1,NW)                                       1781T189
      READ (5,1)(XWH(I),I=1,NW)                                         1781T190
      WRITE (6,9)                                                       1781T191
      WRITE (6,17)(XWH(I),WALLH(I),I=1,NW)                              1781T192
C                                                                       1781T193
C     DU/DY TABLE                                                       1781T194
C                                                                       1781T195
   98 IF(NDU) 985,985,980                                               1781T196
  980 READ (5,1)(DUT(I),I=1,NDU)                                        1781T197
      READ (5,1)(XDUT(I),I=1,NDU)                                       1781T198
      WRITE (6,209)                                                     1781T199
      WRITE (6,17)(XDUT(I),DUT(I),I=1,NDU)                              1781T200
C                                                                       1781T201
C     V(1)                                                              1781T202
C                                                                       1781T203
  985 IF(NV1)99,99,986                                                  1781T204
  986 READ (5,1)(V1T(I),I=1,NV1)                                        1781T205
      READ (5,1)(XV1T(I),I=1,NV1)                                       1781T206
      WRITE (6,217)                                                     1781T207
```

```
      WRITE (6,17)(XV1T(I),V1T(I),I=1,NV1)                              1781T208
C                                                                       1781T209
C     WRITE OUT VELOCITY AND ENTHALPY PROFILES                          1781T210
C                                                                       1781T211
      WRITE (6,218)(U(I),H(I),I=1,LIST)                                 1781T212
  218 FORMAT(31HOVELOCITY AND ENTHALPY PROFILES/19X,1HU,19X,1HH/(2F20.8)1781T213
     1)                                                                 1781T214
C                                                                       1781T215
C     READ TABLES AND/OR SET UP INITIAL PROFILE                         1781T216
C     ----AS FAR AS READ IN                                             1781T217
C                                                                       1781T218
C     TEST CHECK AND ERROR****LIST IS POSITIVE                          1781T219
      LOUP=0                                                            1781T220
      ITTER=0                                                           1781T221
   99 IF(CHECK)260,260,251                                              1781T222
  251 IF(ERROR)260,260,253                                              1781T223
C                                                                       1781T224
C     TEST ERROR***LIST IS ZERO                                         1781T225
C                                                                       1781T226
  252 IF(ERROR)953,953,253                                              1781T227
  253 WRITE (6,207)                                                     1781T228
      ERROR=1.                                                          1781T229
      GO TO 18                                                          1781T230
  260 ERROR=0.0                                                         1781T231
      LP(8)=0                                                           1781T232
      PRESS=TAB(X,LP(1))                                                1781T233
      IF(LP(8)-1)990,101,990                                            1781T234
  101 LDP(8)=0                                                          1781T235
      DPDX=TAB(X,LDP(1))                                                1781T236
      IF(LDP(8)-1) 993,102,993                                          1781T237
  102 LR(8)=0                                                           1781T238
      R=TAB(X,LR(1))                                                    1781T239
      IF(LR(8)-1) 996,122,996                                           1781T240
  122 LDR(8)=0                                                          1781T241
      DRDX=TAB(X,LDR(1))                                                1781T242
      IF(LDR(8)-1) 997,89,997                                           1781T243
   89 LDU(8)=0                                                          1781T244
      DUDYL=TAB(X,LDU(1))                                               1781T245
      IF(LDU(8)-1)  998,87,998                                          1781T246
   87 LV1(8)=0                                                          1781T247
      V(1)=TAB(X,LV1(1))                                                1781T248
      IF(LV1(8)-1)994,88,994                                            1781T249
```

```
   88 DO 95 I=1,200
      YI=I-1
      Y(I)=YI*DY
      IF(I-LIST) 891,891,890
  890 U(I)=U(I-1)+DUDYL*DY
      H(I)=H(LIST)
  891 T(I)=(H(I)-U(I)**2/2.)/CP
      XMU(I)=2.272E-08*T(I)**1.5/(T(I)+198.6)
      RHO(I)=PRESS/(1716.*T(I))
   95 CONTINUE
C
C     COMPUTE INITIAL VALUE OF S
C
      S=SI+RHO(J)*U(J)*XMU(J)*R**2*XI
C
C     LOOP TO SAVE INITIAL PROFILES
C
  953 PRESVE=PRESS
      DPDXSV=DPDX
      RSVE=R
      DRDXSV=DRDX
      DUDYSV=DUDYL
      SSVE=S
      V1SVE=V(1)
      DO 954 I=1,200
      USVE(I)=U(I)
      HSVE(I)=H(I)
      TSVE(I)=T(I)
      XMUSVE(I)=XMU(I)
      RHOSVE(I)=RHO(I)
  954 CONTINUE
C
C     CALCULATE Y DERIVATIVES, V PROFILE, DVDY
C
      NLOOP=0
      NDO=ETAMAX
      NDO1=NDO
   29 DELETA=U(J)*R*DY*XMU(J)**.6/S**.8
      NI=0
      NJ=0
      DVDY(1)=V(1)*(T(2)/T(1)-1.)/DY
```

```
      LOOP=1                                                            1781T291
      TAULY(1)=DPDX                                                     1781T292
      TAUTY(1)=0.                                                       1781T293
      TAULP(1)=0.                                                       1781T294
      TAUTP(1)=0.                                                       1781T295
      DQ(1)=0.                                                          1781T296
      QP(1)=0.                                                          1781T297
      IF(NLOOP)320,320,310                                              1781T298
  310 CONTINUE                                                          1781T299
      NDO=199                                                           1781T300
  320 CONTINUE                                                          1781T301
      DO 40 I=2,NDO                                                     1781T302
      TURBS(I)=.054*(RHO(I)*U(I)*Y(I)/XMU(I))**.833*(U(I)/U(J))**12     1781T303
     1*(T(J)/T(I))**4                                                   1781T304
      DMUDY(I)=(XMU(I+1)-XMU(I-1))/(2.*DY)                              1781T305
      IF(I-2)5000,5000,5001                                             1781T306
 5000 CONTINUE                                                          1781T307
      DUDY(2)=(9.*U(3)-2.*U(4)-6.*U(2))/6./DY                           1781T308
      DHDY(2)=(9.*H(3)-2.*H(4)-6.*H(2)-H(1))/6./DY                      1781T309
      D2UDY(2)=(U(4)+3.*U(3)-9.*U(2))/6./DY**2                          1781T310
      D2HDY(2)=(H(4)+3.*H(3)-9.*H(2)+5.*H(1))/6./DY**2                  1781T311
      GO TO 5002                                                        1781T312
 5001 CONTINUE                                                          1781T313
      DUDY(I)=(U(I+1)-U(I-1))/(2.*DY)                                   1781T314
      DHDY(I)=(H(I+1)-H(I-1))/(2.*DY)                                   1781T315
      D2UDY(I)=(U(I+1)+U(I-1)-2.*U(I))/DY**2                            1781T316
      D2HDY(I)=(H(I+1)+H(I-1)-2.*H(I))/DY**2                            1781T317
 5002 CONTINUE                                                          1781T318
      TAULY(I)=XMU(I)*D2UDY(I)+DMUDY(I)*DUDY(I)                         1781T319
      TAUTY(I)=TURBS(I)*TAULY(I)                                        1781T320
      TAULP(I)=TAULP(I-1)+(TAULY(I-1)+TAULY(I))*DY/2.                   1781T321
      TAUTP(I)=TAUTP(I-1)+(TAUTY(I-1)+TAUTY(I))*DY/2.                   1781T322
   31 ETA(I)=ETA(I-1)+(RHO(I)+RHO(I-1))*DELETA                          1781T323
C                                                                       1781T324
C     ERROR CRITERIA                                                    1781T325
C                                                                       1781T326
      KK=I                                                              1781T327
   35 IF(U(I)+U(J)) 63,37,37                                            1781T328
   37 IF(H(I)) 64,64,41                                                 1781T329
   41 IF(T(I)) 65,65,43                                                 1781T330
   43 IF(X) 430,430,44                                                  1781T331
  430 IF(I-LIST4) 40,435,435                                            1781T332
```

```
  435 K=LIST
      J=LIST4
      GO TO 49
C
C     PROFILE SUCCESS CRITERIA
C
   44 CONTINUE
      IF(NLOOP-1)40,640,45
  640 IF(NI)641,641,5044
  641 IF((1.-U(I-1)/U(I)-DUDYL*DY/U(I))-.00001)647,647,5044
  647 NI=1
 5044 IF(NJ)642,642,5045
  642 IF((1.-H(I-1)/H(I))-.00005)643,643,5045
  643 NJ=1
 5045 NIJ=NI*NJ
      IF(NIJ)40,40,644
  644 NDO1=I+5
      NLOOP=2
      GO TO 40
   45 CONTINUE
      IF(I-NDO1)40,646,646
  646 CONTINUE
      K=I
      J=I
      GO TO 49
   40 CONTINUE
      J=NDO
      K=NDO
C
C     CALCULATE X DERIVATIVES
C
   49 CONTINUE
      NLOOP=1
      DO 4049 I=2,J
      TAUT(I)=TAUTP(I)-TAUTP(J)
      TAUL(I)=TAULP(I)-TAULP(J)
      TAU(I)=TAUL(I)+TAUT(I)
 4049 CONTINUE
      TAU(1)=-TAUTP(J)-TAULP(J)
      DO 3049 I=2,J
      EPS(I)=TAUT(I)/DUDY(I)
```

```
      EPSY(I)=(TURBS(I)*TAULY(I)-EPS(I)*D2UDY(I))/DUDY(I)               1781T374
      DQ(I)=1./PR*((XMU(I)+EPS(I))*D2HDY(I)+(DMUDY(I)+EPSY(I))*         1781T375
     1DHDY(I)-(U(I+1)*TAU(I+1)-U(I-1)*TAU(I-1))/2./DY)                  1781T376
      QP(I)=QP(I-1)+(DQ(I-1)+DQ(I))*DY/2.                               1781T377
      PART1=U(I)**2*(DRDX/R+DPDX/PRESS)                                 1781T378
      PART2=1./RHO(I)/U(I)*(TAULY(I)+TAUTY(I)-DPDX)                     1781T379
      PART3=1./RHO(I)/U(I)*(DQ(I)+(U(I+1)*TAU(I+1)-U(I-1)*TAU(I-1))     1781T380
     1/2./DY)                                                           1781T381
      PART6=U(I)*(2.*V(I-1)/DY+DVDY(I-1))                               1781T382
      V(I)=1./(DUDY(I)-2.*U(I)/DY)*(PART1+U(I)*PART2-U(I)**2/           1781T383
     1(H(I)-U(I)**2/2.)*(PART3-U(I)*PART2)-PART6)                       1781T384
      DVDY(I)=2./DY*(V(I)-V(I-1))-DVDY(I-1)                             1781T385
      DUDX(I)=PART2-V(I)/U(I)*DUDY(I)                                   1781T386
      DHDX(I)=PART3-V(I)/U(I)*DHDY(I)                                   1781T387
 3049 CONTINUE                                                          1781T388
      IF(DS)50,50,3                                                     1781T389
    3 IF(X-STARTX)300,300,50                                            1781T390
  300 I=1                                                               1781T391
    4 IF (ABS(DUDX(I)-U(I)/X)-.1*U(LIST)) 24, 24, 25                    1781T392
   24 I=I+1                                                             1781T393
      IF (I-LIST) 4, 4, 50                                              1781T394
   25 DO 21 I=1,LIST                                                    1781T395
      H(I)=H(I)+DHDX(I)*DS                                              1781T396
      U(I)=(U(I)+DUDX(I)*DS)*(STARTX/(STARTX+DS))                       1781T397
      IF (H(I)) 70,70,71                                                1781T398
   70 DS=DS/2.                                                          1781T399
      LOUP=LOUP+1                                                       1781T400
      IF(LOUP-2)75,75,1007                                              1781T401
C ********** STATEMENT BELOW IS IN ERROR **********                     1781T402
   75 WRITE (6,1015)DS                                                  1781T403
 1015 FORMAT(88H DURING THE STAGNATION SEARCH A NEGATIVE H WAS ENCOUNTER1781T404
     1ED, WE WILL RUN AGAIN USING DS= F8.6)                             1781T405
      GO TO 1003                                                        1781T406
   71 IF(U(I))72,21,21                                                  1781T407
   72 DS=DS/2.                                                          1781T408
      LOUP=LOUP+1                                                       1781T409
      IF(LOUP-2)76,76,1007                                              1781T410
C ********** STATEMENT BELOW IS IN ERROR **********                     1781T411
   76 WRITE (6,1016)DS                                                  1781T412
 1016 FORMAT(88H DURING THE STAGNATION SEARCH A NEGATIVE U WAS ENCOUNTER1781T413
     1ED, WE WILL RUN AGAIN USING DS= F8.6)                             1781T414
      GO TO 1003                                                        1781T415
```

```
   21 CONTINUE
      ITTER=ITTER+1
      IF (ITTER-MITR) 88, 88, 23
C ********** STATEMENT BELOW IS IN ERROR **********
   23 WRITE (6,1005)MITR
 1005 FORMAT(7H AFTER ,I3,94H PROFILES,THE QUANTITY (DUDX-U/X) IS GREATE
     1R IN ABSOLUTE MAGNITUDE THAN ONE PERCENT OF U(LIST))
      GO TO 999
C
C     PROFILE COMPLETED
C
   50 IF(X-XJOT+DX/2.) 108,108,51
  108 IF(X-ENDX+DX/2.) 22,22,51
C
C     COMPUTE Q-WALL,TAU-WALL AND DELTA STAR
C
   51 CONTINUE
      QQW = 0.
      DELSTR=0.
      THETA=0.
      Q(1)=(-QP(J)+U(2)*TAU(2)/2.)/(1.+RHO(1)*V(1)*PR
     1/XMU(1)*DY/2.)/778.
      DO 3150 I=2,J
      Q(I)=(QP(I)-QP(J))/778.
 3150 CONTINUE
      DO 150 I=2,K
      DQQW = (RHO(I)*U(I)*DHDX(I)+RHO(I-1)*U(I-1)*DHDX(I-1)+
     1RHO(I)*V(I)*DHDY(I)+RHO(I-1)*V(I-1)*DHDY(I-1))*.5-
     2((U(I)*TAU(I)-U(I-1)*TAU(I-1))/DY)
      QQW = QQW-DQQW
      DELSTR=DELSTR+1.-(RHO(I)*U(I)+RHO(I-1)*U(I-1))/(2.*RHO(J)*U(J))
      THETA=THETA+(RHO(I)*U(I)              *(1.-U(I)/U(J))
     1+RHO(I-1)*U(I-1)*(1.-U(I-1)/U(J)))/RHO(J)/U(J)
  150 CONTINUE
      QDOTW=Q(1)
      QW=QDOTW
      QQW = QQW*DY/778.
      TAUW=-TAUTP(K)-TAULP(K)
      TAULW=-TAULP(K)
      TAUTW=-TAUTP(K)
      DELSTR=DELSTR*DY
```

```
      THETA=THETA*DY/2.                                                 1781T457
C                                                                       1781T458
C     PRINT VARIABLES                                                   1781T459
C                                                                       1781T460
   55 WRITE (6,10)(TITLE(I),I=1,8),X                                    1781T461
      WRITE (6,11)                                                      1781T462
      WRITE (6,12)(Y(I),U(I),Q(I),DVDY(I),V(I),DUDX(I), T(I),TAU(I),H(I)1781T463
     1 ,ETA(I),DHDX(I),EPS(I),I=1,K)                                    1781T464
      WRITE (6,206)           QW,TAULW,TAUTW,DELSTR,THETA,ITTER         1781T465
      WRITE (6,2006)RHO(LIST),DPDX                                      1781T466
 2006 FORMAT(7H RHO = E14.8,5X,7HDPDX = E14.8)                          1781T467
      WRITE (6,2060)QQW                                                 1781T468
 2060 FORMAT(25H QQW FOR THIS PROFILE IS E12.5)                         1781T469
      WRITE(6,2061)S                                                    1781T470
 2061 FORMAT(5H S = ,E12.5)                                             1781T471
      IF(X-XJOTT(LX)+DX/2.) 52,52,53                                    1781T472
   52 XJOT=XJOT+DJOT(LX)                                                1781T473
      GO TO 54                                                          1781T474
   53 LX=LX+1                                                           1781T475
      XJOT=XJOT+DJOT(LX)                                                1781T476
   54 IF(X-ENDX+DX/2.) 22,22,18                                         1781T477
C                                                                       1781T478
C     STEP UP X AND READ NEW X-DEPENDENT VARIABLES                      1781T479
C                                                                       1781T480
   22 TIMES=TIMES+1.                                                    1781T481
      X=STARTX+TIMES*DX                                                 1781T482
      LP(8)=0                                                           1781T483
      PRESS=TAB(X,LP(1))                                                1781T484
      IF(LP(8)-1) 990,104,990                                           1781T485
  104 LDP(8)=0                                                          1781T486
      DPDX=TAB(X,LDP(1))                                                1781T487
      IF(LDP(8)-1) 993,110,993                                          1781T488
  110 LR(8)=0                                                           1781T489
      R=TAB(X,LR(1))                                                    1781T490
      IF(LR(8)-1) 996,123,996                                           1781T491
  123 LDR(8)=0                                                          1781T492
      DRDX=TAB(X,LDR(1))                                                1781T493
      IF(LDR(8)-1) 997,90,997                                           1781T494
   90 LH(8)=0                                                           1781T495
      H(1)=TAB(X,LH(1))                                                 1781T496
      IF(LH(8)-1) 995,109,995                                           1781T497
  109 LDU(8)=0                                                          1781T498
```

```
      DUDYL=TAB(X,LDU(1))                                               1781T499
      IF(LDU(8)-1) 998,1079,998                                         1781T500
 1079 LV1(8)=0                                                          1781T501
      V(1)=TAB(X,LV1(1))                                                1781T502
      IF(LV1(8)-1)994,1080,994                                          1781T503
C                                                                       1781T504
C     INTEGRATION LOOP                                                  1781T505
C                                                                       1781T506
C     ----MOVE X-DEPENDENT VARIABLES TO NEW PROFILE                     1781T507
 1080 S=S+RHO(J)*XMU(J)*U(J)*R**2*DX                                    1781T508
      DO 124 I=1,200                                                    1781T509
      IF (I-1) 1084, 1084, 1081                                         1781T510
 1081 IF (I-K) 1082, 1082, 1083                                         1781T511
 1082 H(I)=H(I)+DHDX(I)*DX                                              1781T512
      U(I)=U(I)+DUDX(I)*DX                                              1781T513
      GO TO 1084                                                        1781T514
 1083 H(I) = H(K)                                                       1781T515
      U(I) = U(I-1) + DUDYL*DY                                          1781T516
 1084 T(I)=(H(I)-U(I)**2/2.)/CP                                         1781T517
      IF(T(I))65,65,1085                                                1781T518
 1085 XMU(I)=2.272E-08*T(I)**1.5/(T(I)+198.6)                           1781T519
      RHO(I)=PRESS/(1716.*T(I))                                         1781T520
  124 CONTINUE                                                          1781T521
      GO TO 29                                                          1781T522
C                                                                       1781T523
C     ERROR INDICATORS                                                  1781T524
C                                                                       1781T525
   63 WRITE (6,14)                                                      1781T526
      GO TO 1000                                                        1781T527
   64 WRITE (6,15)                                                      1781T528
      GO TO 1000                                                        1781T529
   65 WRITE (6,13)                                                      1781T530
      GO TO 1000                                                        1781T531
  990 WRITE (6,2)                                                       1781T532
      GO TO 999                                                         1781T533
  993 WRITE (6,5)                                                       1781T534
      GO TO 999                                                         1781T535
  994 WRITE (6,216)                                                     1781T536
      GO TO 999                                                         1781T537
  995 WRITE (6,6)                                                       1781T538
      GO TO 999                                                         1781T539
```

```
  996 WRITE (6,19)                                                        1781T540
      GO TO 999                                                           1781T541
  997 WRITE (6,200)                                                       1781T542
      GO TO 999                                                           1781T543
  998 WRITE (6,210)                                                       1781T544
  999 WRITE (6,10)(TITLE(I),I=1,8),X                                      1781T545
      WRITE (6,11)                                                        1781T546
      WRITE (6,12)(Y(I),U(I),Q(I),DVDY(I),V(I),DUDX(I), T(I),TAU(I),H(I)1781T547
     1,ETA(I),DHDX(I),EPS(I),I=1,KK)                                      1781T548
 1007 ERROR=1.0                                                           1781T549
      GO TO 18                                                            1781T550
 1000 WRITE (6,10)(TITLE(I),I=1,8),X                                      1781T551
      WRITE (6,11)                                                        1781T552
      WRITE (6,12)(Y(I),U(I),Q(I),DVDY(I),V(I),DUDX(I), T(I),TAU(I),H(I)1781T553
     1,ETA(I),DHDX(I),EPS(I),I=1,KK)                                      1781T554
C                                                                         1781T555
C     DECREASE DX AND RESTORE INITIAL PROFILES                            1781T556
C                                                                         1781T557
      IF(NTRY-3)1001,1007,1007                                            1781T558
 1001 NTRY=NTRY+1                                                         1781T559
      STOP                                                                1781T560
      X=STARTX                                                            1781T561
      XJOT=STARTX                                                         1781T562
      LX=1                                                                1781T563
      TIMES=0.                                                            1781T564
      DX=DX/2.                                                            1781T565
      DS=0.                                                               1781T566
      WRITE (6,16)NTRY,DX                                                 1781T567
      IF(LIST)1003,1003,1002                                              1781T568
C                                                                         1781T569
C     LOOP TO RESTORE INITIAL PROFILES                                    1781T570
C                                                                         1781T571
 1002 J=LIST4                                                             1781T572
 1003 PRESS=PRESVE                                                        1781T573
      DPDX=DPDXSV                                                         1781T574
      R=RSVE                                                              1781T575
      DRDX=DRDXSV                                                         1781T576
      DUDYL=DUDYSV                                                        1781T577
      S=SSVE                                                              1781T578
      V(1)=V1SVE                                                          1781T579
      DO 1004 I=1,200                                                     1781T580
      U(I)=USVE(I)                                                        1781T581
```

```
      H(I)=HSVE(I)                                                      1781T582
      T(I)=TSVE(I)                                                      1781T583
      XMU(I)=XMUSVE(I)                                                  1781T584
      RHO(I)=RHOSVE(I)                                                  1781T585
 1004 CONTINUE                                                          1781T586
      GO TO 29                                                          1781T587
      END                                                               1781T588

                                                              000588
```

REFERENCES

1. Hanks, R. A.: and Savage, R. T.: Thermal Design Methods for Recoverable Launch Vehicles with Consideration of Arbitrary Wall Temperature and Surface Conditions, NASA CR-74714, August 1965.

2. Nagel, A. L.: Fitzsimmons, H. D.; and Doyle, L. B.: Analysis of Hypersonic Pressure and Heat Transfer Tests on Delta Wings with Laminar and Turbulent Boundary Layers, NASA CR-535, August 1966.

3. Thomas, A. C.: Perlbachs, A.; and Nagel, A. L.: Advanced Reentry Systems Heat Transfer Manual for Hypersonic Flight, AFFDL-TR-65-195, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, January 1966.

4. Hansen, C. F.: Approximations for the Thermodynamic and Transport Properties of High Temperature Air, NACA TN-4150, March 1958.

5. Hilsenrath, J.: and Beckett, C. W.: Thermodynamic Properties of Argon-Free Air (0.78847 N_2, 0.21153 O_2) to 15,000 °K, NBS Report No. 3991, July 1955.

6. Feldman, S.: Hypersonic Gas Dynamic Charts for Equilibrium Air, AVCO Research Report No. 40, January 1957.

7. Logan, J.; and Treanor, C.: Tables of Thermodynamic Properties of Air from 3,000 to 10,000 °K. Cornell Aeronautical Laboratory, Inc., January 1957.

8. Ames Research Staff: Equations, Tables, and Charts for Compressible Flow, NACA Report 1135, 1953.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified).

1. ORIGINATING ACTIVITY (Corporate author) The Boeing Company P. O. Box 3868 Aerospace Group Seattle, Washington Space Division—Kent Facility 98124	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE
INVESTIGATION OF TURBULENT HEAT TRANSFER AT HYPERSONIC SPEEDS
Volume III. The Laminar-Turbulent $\rho_r \mu_r$ Momentum Integral
and Turbulent Nonsimilar Boundary Layer Computer Programs

4. DESCRIPTIVE NOTES (Types of report and inclusive dates)
Final Report, March 1965 to March 1967

5. AUTHORS (First name, middle initial, last name)
Savage, Richard T.
Jaeck, Carl L.
Mitchell, James R.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1967	222	8

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBERS
AF33(615)-2372 b. Project No. 1366 c. Task No. 136607 d. BPSN 5(61136-62405334)	D2-113531-3 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFFDL-TR-67-144, Volume III

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale;
its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Air Force Flight Dynamics Laboratory Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433

13. ABSTRACT
This report presents a combined analytical and experimental investigation of turbulent heat transfer on basic and composite configurations at hypersonic speeds. The analytical results are presented in Volume I, the experimental results, including data-theory comparisons, are presented in Volume II, and computer programs incorporating the analytical methods described herein are presented in Volume III of this report.

The two heat-transfer prediction methods programmed are the laminar-turbulent $\rho_r \mu_r$ momentum integral method and the turbulent nonsimilar method. This volume of the report describes the numerical method and presents flow charts, program listings, input forms, and a description of the output for each program. The programs are written in Fortran IV language for operation on the IBM 7094.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Nonsimilar Boundary Layer						
Momentum Integral						
Aerodynamic Heating						
Laminar Flow						
Turbulent Flow						